TOWARDS 3-DIMENSIONAL REWRITING THEORY

SAMUEL MIMRAM

CEA, LIST, LMeASI, 91191 Gif-sur-Yvette, France.
E-mail address: samuel.mimram@cea.fr

ABSTRACT. String rewriting systems have proved very useful to study monoids. In good cases, they give finite presentations of monoids, allowing computations on those and their manipulation by a computer. Even better, when the presentation is confluent and terminating, they provide one with a notion of canonical representative of the elements of the presented monoid. Polygraphs are a higher-dimensional generalization of this notion of presentation, from the setting of monoids to the much more general setting of n-categories. One of the main purposes of this article is to give a progressive introduction to the notion of *higher-dimensional rewriting system* provided by polygraphs, and describe its links with classical rewriting theory, string and term rewriting systems in particular. After introducing the general setting, we will be interested in proving local confluence for polygraphs presenting 2-categories and introduce a framework in which a finite 3-dimensional rewriting system admits a finite number of critical pairs.

Recent developments in category theory have established higher-dimensional categories as a fundamental theoretical setting in order to study situations arising in various areas of mathematics, physics and computer science. A nice survey of these can be found in [2], explaining how the use of category theory enables one to unify these apparently unrelated fields of science, by revealing that their intrinsic algebraic structures are in fact closely connected. In the last decade, higher-dimensional categories have therefore emerged as a tool of everyday use for many scientists. The motivation behind the concept of higher dimensions here is that, in order to have a fine-grained understanding of the algebraic structures at stake, one should not only consider morphisms between objects involved, but also morphisms between morphisms (i.e. 2-dimensional morphisms), morphisms between morphisms between morphisms (i.e. 3-dimensional morphisms), and so on. For example, the starting point of algebraic topology [16] is that one should not consider points and paths between them in topological spaces, but also homotopies between paths, and can be refined by also considering homotopies between homotopies and so on.

The categorical structures considered nowadays are thus becoming more and more complex, which enables them to capture many details, but the proofs are becoming more and more complicated too, and we are facing the urge for new tools, both of a theoretical and practical nature, in order to make them easier and more manageable. In particular, many proofs of even conceptually simple facts involve showing the commutativity of diagrams

which are big both in size and dimension: for example, the proof that a commutative strong monad on a symmetric monoidal category (which is a particular 3-category) is the same notion as a symmetric monoidal monad involves checking the commutativity of many large diagrams as detailed in the appendix of [9], as another example, the very definition of tricategories (a weak form of 3-category) as well as the associated morphisms takes up pages, not to mention the associated coherence theorem [8, 15]. These are, among many other, striking examples of the fact that we are slowly approaching the size limits of diagrams that can be computed by hand in a decent time, or decently written on paper. Moreover, these computations do not really constitute the conceptually interesting part of the work, and are often considered as "routine checks" because they are very repetitive and systematic. . . which is a good point from a computer scientist's point of view: we have hope to be able to develop software to check and automate them!

With this goal in mind, rewriting theory is a natural candidate for providing powerful tools in order to study these algebraic structures. Its uses have proven very useful for example to construct and manipulate *presentations* of monoids, which are descriptions of monoids by generators and relations: a string rewriting system is simply a presentation of a monoid where the relations are oriented so that they form rewriting rules. When the rewriting system is normalizing (which is in particular the case when it is locally confluent and terminating), it provides one with a notion of canonical representative of words modulo the relations, which turns out to be crucial to manipulate the presentations. Starting from this point, Burroni has introduced the notion of *polygraph* [5], rediscovering Street and Power's *computads* [36, 32], generalizing the notion of presentation from monoids to higher-dimensional categories, thus providing us with a good notion of *higher-dimensional rewriting system*, which is able to describe and manipulate free n-categories modulo equations on n-cells. This was later on used intensively by Lafont in order to construct presentations of various monoidal categories [22] and further studied by Guiraud [10, 11, 12, 13] and Malbos [14].

However, much more than a mere adaptation of the well-known techniques of rewriting theory is needed here. In particular, Lafont discovered a very interesting fact: contrarily to string or term rewriting systems, these generalized rewriting systems can give rise to an *infinite* number of critical pairs, even when they are *finite*! In this article, we will be interested in extending techniques for computing critical pairs (we will not detail techniques for showing termination [11]). Its main contribution is to address this problem, in the case of 3-dimensional rewriting systems, by generalizing the notion of critical pair in order to recover a finite number of critical pairs for finite rewriting systems. The present work constitutes a first step in the direction of generalizing rewriting theory to higher dimensions, but a lot of efforts remain to be done in order to achieve the task.

Unfortunately, the idea of higher-dimensional rewriting system does not seem to have spread very much among the community of rewriting theory. We believe this is partly due to the fact that its fundamental concepts are scattered in various papers, which sometimes require a strong categorical background, or are even sometimes considered as "folklore". In order to address this, we have done our best to gradually introduce the topic, in a way accessible to people familiar with rewriting theory [1, 37] and basic (1-dimensional) category theory [26], and tried to write an introductory article on the subject. We have favored the exposition instead of handling directly the most technical matters, trying to provide intuitions about the structures involved, and giving bibliographical references for

the reader interested in details concerning a particular point. Likewise, a completely formal and more abstract presentation of all the contributions of the author in the article – as well as the other developments exposed by the author in [30] – shall be presented in a subsequent article [31]. We feel that an introductory paper such as this one was really necessary first, in order for the new developments to be accessible to a broad audience.

We first recall in Section 1 the notion of higher-dimensional rewriting system provided by polygraphs and explain its links with string and term rewriting systems. The following Section 2 describes how these rewriting systems provide a good notion of presentation of n-categories. Finally, we explain in Section 3 why a finite 3-dimensional rewriting system can give rise to an infinite number of critical pairs and introduce a theoretical setting to overcome this issue, and conclude in Section 4.
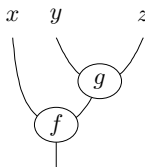
1. Towards higher-dimensional rewriting systems

In order to generalize the notion of rewriting system to higher dimensions, one should follow the principles established by topology and category theory: a 0-dimensional rewriting system should be a set of "points" and an $(n+1)$-dimensional rewriting system should consist of rules which *rewrite rewriting paths* in an n-dimensional rewriting system. Informally, one should think of the well-known situation arising in the study of λ-calculus: if we consider the λ-terms modulo α-conversion as 0-dimensional terms, the rules for β-reduction would play the role of a 1-dimensional rewriting system, and the rules for standardization [7], which rewrite any reduction sequence into a "standard" one, would play the role of a 2-dimensional rewriting system because they rewrite rewriting paths.

The idea is quite appealing, but it is not clear at first how the classical rewriting frameworks should fit in this picture, and one should first try to understand in a uniform way the two major examples of rewriting theory: *string rewriting systems* and *term rewriting systems*. Namely, string rewriting systems should intuitively be part of 2-dimensional rewriting systems because they rewrite words which can be considered as 1-dimensional objects: a word abc can be seen geometrically as a path

$$\xrightarrow{\;a\;} \xrightarrow{\;b\;} \xrightarrow{\;c\;}$$

Similarly, term rewriting systems should be part of 3-dimensional rewriting systems because they rewrite terms which can be considered as 2-dimensional objects, if we picture a term as a 2-dimensional diagram, using the representation often used for sharing graphs: for example, the term $f(x, g(y, z))$ can be represented by

However, this intuition does not give precise directions about how a 4-dimensional rewriting system should be defined for example. It turns out that generalizing a bit the settings of string and term rewriting systems reveals an inductive pattern in the definition of these rewriting frameworks, whose discovery motivated the definition of *polygraphs*, which can be

considered as the good notion of higher-dimensional rewriting system. We shall begin by explaining this in details.

1.1. **Graphs and 1-dimensional rewriting systems.**

1.1.1. *Definition.* Rewriting systems in dimension 1 should be defined in the most simple way. The fact that a topological space of dimension 0 is a point suggests that we should define a *0-signature* simply as a set E_0, whose elements are called (0-dimensional) *terms*. A 1-dimensional rewriting system on the signature E_0 is a set of *rules* rewriting a term into another term. It therefore consists of a set E_1, whose elements are the *rules*, together with two functions $s, t : E_1 \to E_0$ which to every rule $r \in E_1$ associate its *source* $s(r)$ and its *target* $t(r)$. In other words, a 1-rewriting system is a diagram

$$E_0 \underset{t}{\overset{s}{\longleftarrow\joinrel\longleftarrow}} E_1 \tag{1.1}$$

in the category **Set** (the category whose objects are sets and morphisms are functions). We sometimes write

$$r \quad : \quad A \quad \to \quad B$$

to indicate that r is a rule with A as source and B as target and say that A *one-step rewrites* by r to B. We simply write $A \to B$ when there exists a rule $r : A \to B$, write $\to^*$ for the transitive closure of the relation $\to$, and say that a term A *rewrites* to a term B when $A \to^* B$.

1.1.2. *The free category generated by a rewriting system.* Notice that a rewriting system of the form (1.1) can also be seen as a graph with E_0 as set of vertices and E_1 as set of edges, the functions s and t indicating the source and the target of an edge. Such a graph generates a *free category* $\mathcal{C}$ which is the smallest category with E_0 as set of objects and containing all the edges $r : A \to B$ in E_1 as morphisms. We write E_1^* for the set of morphisms of this category, and $s^*, t^* : E_1^* \to E_0$ for the functions which to every morphism of this category associate its source and target respectively. Since the set E_1^* contains the set E_1, there is an injection $i_1 : E_1 \to E_1^*$. Moreover, since $\mathcal{C}$ is the free category generated by the graph (1.1), E_1^* is the smallest set containing E_1 and closed under identities and composition: E_1^* is the set of *paths* in the graph, with concatenation as composition and empty paths as identities, with the obvious source and target functions s_1^* and t_1^*. The data generated by this free construction can be summarized by a diagram

$$
\begin{array}{ccc}
 & & E_1 \\
 & {\overset{s}{\swarrow}\nearrow} & \downarrow {\scriptstyle i_1} \\
 & {\overset{s^*}{\swarrow} t} & \\
E_0 & \underset{t^*}{\overset{}{\longleftarrow\joinrel\longleftarrow}} & E_1^*
\end{array}
$$

in **Set**, together with the composition morphism $E_1^* \times_{E_0} E_1^* \to E_1^*$ which to every pair of composable morphisms associate their composition (i.e. the concatenation of the two paths) and the identity morphism $E_0 \to E_1^*$ which to every object in E_0 associates an identity morphism (i.e. the empty path on this object): these two last operations define a structure of category on the graph $E_0 \underset{t^*}{\overset{s^*}{\longleftarrow\joinrel\longleftarrow}} E_1^*$ (a category is a graph together with composition

and identity operations satisfying axioms). Moreover, the diagram above "commutes" in the sense that

$$s^* \circ i = s \qquad \text{and} \qquad t^* \circ i = t$$

which expresses the fact that if $r : A \to B$ is an arrow in the graph, then it has the same source and target if we see it as a path of length 1 in the graph. Notice that, by definition, a term A rewrites to a term B when there exists a path from A to B in the graph, i.e. when there exists a morphism from A to B in the generated category: the morphisms of the freely generated category can be seen as rewriting sequences.

Example 1.1. Let us give examples of categories generated by graphs.

(1) Consider the graph with two edges $E_0 = A, B$ and two arrows $f : A \to B$ and $g : B \to A$

$$A \underset{g}{\overset{f}{\rightleftarrows}} B$$
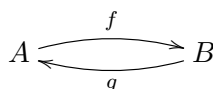
i.e. formally

$$E_0 = \{A, B\} \qquad E_1 = \{f, g\} \qquad s(f) = t(g) = A \qquad t(f) = s(g) = B$$

It generates a category which has the words of the form $(fg)^*$ as morphisms from A to A (where the star denotes the Kleene star), $(fg)^*f$ from A to B, $g(fg)^*$ from B to A, and $(gf)^*$ from B to B, i.e. formally $E_1^* = (fg)^* \uplus g(fg)^* \uplus (fg)^*f \uplus (gf)^*$.

(2) The following graph with one vertex and one arrow

$$* \overset{}{\underset{}{\circlearrowright}} 1$$

generates a category with one object isomorphic to the category whose morphisms are natural integers $\mathbb{N}$ with addition as composition and 0 as identity.

(3) More generally, given a set Σ, the graph G_Σ with one object $*$ and Σ as set of arrows (all from $*$ to $*$) generates a category isomorphic to the category with one object and the free monoid Σ^* on Σ as set of morphisms, with concatenation as composition and empty word as identity. In other words, the category generated by G_Σ is essentially the same as the free monoid Σ^* generated by Σ.

1.2. String and 2-dimensional rewriting system.

1.2.1. *String rewriting systems.* Suppose given an *alphabet* E_1. We write E_1^* for the free monoid generated by E_1 and $i_1 : E_1 \to E_1^*$ for the injection sending a letter of the alphabet to the corresponding word with only one letter. A *string rewriting system* R on the alphabet E_1 is usually defined as a relation $R \subseteq E_1^* \times E_1^*$ over the free monoid generated by E_1. One says that a word u *one-step rewrites* to a word v, what we write $u \Rightarrow v$, when there exists two words $w_1, w_2 \in E_1^*$ such that

$$u = w_1 u' w_2 \qquad v = w_1 v' w_2 \quad \text{and} \quad (u', v') \in R \tag{1.2}$$

If we write $\Rightarrow^*$ for the reflexive and transitive closure of the relation $\Rightarrow$, a word u *rewrites* to a word v when $u \Rightarrow^* v$.

In order to make the definition closer to the one given in previous section, we can make a small generalization and allow to have two distinct rules both rewriting the same word u to the same word v, by giving *names* to the rules. Such a string rewriting system can

be formalized as a set E_2 whose elements are the (names of) the rules together with two functions $s_1, t_1 : E_2 \to E_1^*$ indicating respectively the *source* and *target* of a rule:

$$
\begin{array}{ccc}
E_1 & & E_2 \\
& \searchrow s_1 & \\
i_1 \downarrow & & \swarrow t_1 \\
E_1^* & &
\end{array}
$$

and again we write $\rho : u \Rightarrow v$ to indicate that $\rho \in E_2$ is a rule with $s_1(\rho) = u$ and $t_1(\rho) = v$.

Example 1.2. One can for example consider the rewriting system on an alphabet with two letters a, b and a rule $ba \Rightarrow ab$, which would be formalized by

$$ E_1 = \{a, b\} \qquad E_2 = \{\rho\} \qquad s_1(\rho) = b \otimes a \qquad t_1(\rho) = a \otimes b $$

where $\otimes$ denotes the concatenation operation on words.

More details and examples of uses of such rewriting systems are given in Section 2.1.1.

1.2.2. *2-dimensional rewriting systems.* Remember that our general plan is to express 2-dimensional rewriting systems as rewriting on 1-dimensional rewriting paths. In order to do so, one should have in mind that a set is "the same as" a graph with one vertex. Formally, there exist two functors

$$ \mathbf{Set} \xrightleftharpoons[G]{F} \mathbf{Graph}_* $$

which are inverse one of each other, where $\mathbf{Graph}_*$ is the full subcategory of $\mathbf{Graph}$ (the category of graphs) whose objects are graphs with only one vertex: the functor F sends a set Σ to the graph with one vertex $*$ and Σ as set of arrows (all going from $*$ to $*$) and the functor G sends a graph with one vertex to its set of edges. Similarly, a monoid is "the same as" a category with one object, the multiplication and unit of a monoid corresponding to the composition and identity of the category. For instance, we have explained in the last point of Example 1.1 that the free monoid generated by a set is, with respect to this point of view, "the same as" the free category generated by the graph corresponding to the set.

   Therefore, instead of having a mere set as alphabet, one might equivalently declare that an alphabet is a 1-dimensional rewriting system

$$ E_0 \xleftarrow[t_0]{s_0} E_1 \tag{1.3} $$

as defined in Section 1.2, with the set E_0 being reduced to one element. Again, this does not bring more information than the set E_1 itself, since the set E_0 is uniquely defined up to a canonical isomorphism (it contains only one object) and the functions $s_0, t_0 : E_1 \to E_0$ are uniquely defined because E_0 is terminal in $\mathbf{Set}$. If we write E_1^* for the set of morphisms of the category generated by the graph (1.3) as in previous section, a 2-dimensional rewriting system can thus be defined as a diagram

$$
\begin{array}{ccccc}
& & E_1 & & E_2 \\
& s_0 \swarrow & \downarrow i_1 & s_1 \swarrow & \\
& \swarrow & & \swarrow t_1 & \\
E_0 & \xleftarrow[t_0^*]{s_0^* \, t_0} & E_1^* & &
\end{array}
\tag{1.4}
$$

in **Set**, with E_0 containing only one element, together with a structure of category on the graph

$$E_0 \underset{t_0^*}{\overset{s_0^*}{\longleftarrow}} E_1^* \tag{1.5}$$

It turns out that the supposition that the set E_0 is reduced to one element is not necessary in order to proceed with the usual developments of string rewriting theory. We thus drop this condition in the following and suppose that an alphabet for a 2-dimensional rewriting system is any 1-dimensional rewriting system. This amounts to generalizing string rewriting systems to the case where the letters $a : A \to B$ are typed (with A as source and B as target), and to consider only *composable* string of letters as words (the words are thus also typed). In order for substitution to be well-defined, we have to suppose that a word $u : A \to B$ can only be rewritten to a word $v : A' \to B'$ of the same type (i.e. $A' = A$ and $B' = B$). It is enough to suppose that this property is satisfied for the rules, which amounts to impose the further axiom that in any 2-dimensional rewriting system (1.4), the equations

$$s_0^* \circ s_1 = s_0^* \circ t_1 \qquad \text{and} \qquad t_0^* \circ s_1 = t_0^* \circ t_1$$

are satisfied (these are sometimes called the *globular equations*). An example of such a rewriting system is given in Section 1.2.5.

1.2.3. *The free 2-category generated by a rewriting system.* In the same way that a 1-dimensional rewriting system freely generates a 1-category (i.e. a category), a 2-dimensional rewriting system freely generates a 2-category: it is the smallest 2-category with (1.5) as underlying category and the containing the elements of E_2 as 2-cells. We recall that the notion of 2-category is defined as follows.

Definition 1.3. A *2-category* $\mathcal{C}$ consists of the following data.
- A class $\mathcal{C}_0$ of *0-cells*.
- A category $\mathcal{C}(A, B)$ for every pair of 0-cells A and B. Its objects $f : A \to B$ are called *1-cells*, its morphisms $\alpha : f \Rightarrow g : A \to B$ are called *2-cells*, composition is written $\circ$ and called *vertical composition*, and identities are called *vertical identities*.
- A functor $\otimes_{A,B,C} : \mathcal{C}(A, B) \times \mathcal{C}(B, C) \to \mathcal{C}(A, C)$ for every objects A, B and C called *horizontal composition* (we will drop the subscripts of $\otimes$ in the following).
- A 1-cell $\mathrm{id}_A : A \to A$ for every object A called *vertical identity*.

These should be such that the following properties are satisfied.
- Horizontal composition is associative: for every 0-cells A, B, C and D, for every 1-cells $f, f' : A \to B$, $g, g' : B \to C$ and $h, h' : C \to D$, for every 2-cells $\alpha : f \Rightarrow f'$, $\beta : g \Rightarrow g'$ and $\gamma : h \Rightarrow h'$,

$$(f \otimes g) \otimes h = f \otimes (g \otimes h) \quad (\alpha \otimes \beta) \otimes \gamma = \alpha \otimes (\beta \otimes \gamma) \quad (f' \otimes g') \otimes h' = f' \otimes (g' \otimes h')$$

- Horizontal identities are neutral elements for horizontal composition: for every 0-cells A and B, for every 1-cells $f, f' : A \to B$, for every 2-cell $\alpha : f \Rightarrow f'$,

$$\mathrm{id}_A \otimes f = f = f \otimes \mathrm{id}_B \quad \mathrm{id}_{\mathrm{id}_A} \otimes \alpha = \alpha = \alpha \otimes \mathrm{id}_{\mathrm{id}_B} \quad \mathrm{id}_A \otimes f' = f' = f' \otimes \mathrm{id}_B$$

If we write E_2^* for the set of 2-cells of the 2-category generated by the 2-rewriting system (1.4), $i_2 : E_2 \to E_2^*$ for the canonical injection expressing the inclusion of E_2 into E_2^*

and $s_1^*, t_1^* : E_2 \to E_1^*$, we thus get a diagram

$$
\begin{array}{c}
\end{array}
$$

in **Set**, together with a structure of 2-category on the 2-graph

$$
E_0 \xleftarrow[t_0^*]{s_0^*} E_1^* \xleftarrow[t_1^*]{s_1^*} E_2^*
$$

which commutes in the sense that

$$
s_1^* \circ i_2 = s_1 \qquad \text{and} \qquad t_1^* \circ i_2 = t_1
$$

In the same way as previously, the 2-cells of this 2-category correspond to the rewriting paths: a (typed) word $u : A \to B$ rewrites to a (typed) word $w : A \to B$ if and only if there exists a 2-cell $\alpha : u \Rightarrow v : A \to B$. In particular, if there is only one 0-cell, this notion of rewriting corresponds the usual notion of rewriting for string rewriting systems. As previously, notice that a 1-cell is rewritten into a *parallel* 1-cell, i.e. a 1-cell having the same source and same target.

1.2.4. *A diagrammatic notation for 2-cells.* Before going on describing how to incorporate string rewriting systems in our framework, we shall try to understand a bit more 2-categories, and the previous construction.

Given a 2-category $\mathcal{C}$, a 2-cell $\alpha : u \Rightarrow v : A \to B$ is often written diagrammatically as

$$
A \underset{v}{\overset{u}{\Longrightarrow}} \!\!{}_{\alpha\Downarrow}\; B
$$

With this notation, vertical composition corresponds to vertical juxtaposition: the composite $\beta \circ \alpha : u \Rightarrow w : A \to B$ of a pair of 2-cells $\alpha : u \Rightarrow v : A \to B$ and $\beta : v \Rightarrow w : A \to B$ is denoted by

$$
A \;\; \overset{u}{\underset{w}{\Longrightarrow}} \;\; B \tag{1.6}
$$

and the horizontal composition $\alpha \otimes \beta : (u \otimes v) \Rightarrow (u' \otimes v') : A \to C$ of a pair of 2-cells $\alpha : u \Rightarrow u' : A \to B$ and $\beta : v \Rightarrow v' : B \to C$ corresponds diagrammatically to horizontal juxtaposition

$$
A \underset{u'}{\overset{u}{\Longrightarrow}}{}_{\alpha\Downarrow}\; B \; \underset{v'}{\overset{v}{\Longrightarrow}}{}_{\beta\Downarrow}\; C \tag{1.7}
$$

For every 1-cell $u : A \to B$, a vertical identity 2-cell

$$
A \underset{u}{\overset{u}{\Longrightarrow}}{}_{\mathrm{id}_f\Downarrow}\; B
$$

should be given which acts as neutral element for vertical composition.

The functoriality of the horizontal composition implies a last fundamental identity called the *exchange law*. Given four 2-cells α, β, α' and β' as in the diagram

$$A \xrightarrow[\substack{\alpha\Downarrow \\ v \\ \beta\Downarrow}]{u} B \xrightarrow[\substack{\alpha'\Downarrow \\ v' \\ \beta'\Downarrow}]{u'} C \tag{1.8}$$

there are two ways to compose them altogether: either first vertically and then horizontally or the converse. The exchange law which is satisfied in any 2-category states that these two composites are equal:

$$(\beta \circ \alpha) \otimes (\beta' \circ \alpha') \quad = \quad (\beta \otimes \beta') \circ (\alpha \otimes \alpha')$$

Thanks to this, the diagram (1.8) is not ambiguous. This law can equivalently be reformulated slightly differently as the *Godement law*: given two 2-cells α and β as in (1.7), the following identity is always satisfied:

$$(\mathrm{id}_{u'} \otimes \beta) \circ (\alpha \otimes \mathrm{id}_v) \quad = \quad (\alpha \otimes \mathrm{id}_{v'}) \circ (\mathrm{id}_u \otimes \beta) \tag{1.9}$$

1.2.5. *An example.* Consider a 2-dimensional rewriting system of the form (1.4) with

$$E_0 = \{A, B\} \qquad E_1 = \{f, g\} \qquad E_2 = \{\rho\}$$

and source and target map defined by

$$s_0(f) = t_0(g) = A \qquad t_0(f) = s_0(g) = B \qquad s_1(\rho) = f \otimes g \otimes f \otimes g \qquad t_1(\rho) = f \otimes g$$

i.e. using the previously introduced notations

$$f : A \to B \qquad g : B \to A \qquad \rho : f \otimes g \otimes f \otimes g \Rightarrow f \otimes g : A \to B$$

In particular, the 1-rewriting system playing the role of signature for this 2-rewriting system is the one given in first point of Example 1.1, and the $\otimes$ operation denotes the composition of the category it generates.

As explained earlier, A and B should be seen as types for words, f and g should be seen as the letters of the signature and the $\otimes$ operation as the concatenation for words (we thus sometimes omit it and write fg instead of $f \otimes g$). These letters are typed and one only considers words obtained by composing compatible letters; for example $f \otimes g \otimes f$ is a word of type $A \to B$, graphically represented as the path

$$A \xrightarrow{f} B \xrightarrow{g} A \xrightarrow{f} B$$

whereas $f \otimes f$ does not make sense. Finally, ρ is a rewriting rule, which we may represent as a 2-cell

$$\begin{array}{c} B \xrightarrow{g} A \xrightarrow{f} B \\ f \nearrow \qquad \rho\Downarrow \qquad \searrow g \\ A \xrightarrow{f} B \xrightarrow{g} A \end{array} \tag{1.10}$$

which transforms a word into another.

The rewriting paths are the morphisms in the 2-category generated by the 2-rewriting system: its 2-cells are formal vertical and horizontal composites of rules quotiented by the axioms of 2-categories. Vertical compositions correspond to sequences of rewriting: if a word u rewrites to a word v, which is witnessed by a morphism $\alpha : u \Rightarrow v$, and the word v

rewrites to a word w, which is witnessed by a morphism $\beta : v \Rightarrow w$, then the composite morphism $\beta \circ \alpha : u \Rightarrow w$ witnesses the fact that u rewrites to w, which is represented diagrammatically in (1.6). Horizontal compositions correspond similarly to rewritings done in parallel: if a word u rewrites to u' by $\alpha : u \Rightarrow u'$ and a word v rewrites to v' by $\beta : v \Rightarrow v'$ then $u \otimes v$ rewrites to $u' \otimes v'$ by $\alpha \otimes \beta$, as shown in (1.7). For example, the usual condition for one-step rewrites (1.2) can be recovered as follows: the generator ρ rewrites $fgfg$ into fg, so, given two words $u : C \to A$ and $v : A \to D$ (where C and D are either A or B), the word $ufgfgv$ rewrites to $ufgv$, which is witnessed by the morphism

$$\mathrm{id}_u \otimes \rho \otimes \mathrm{id}_v \quad : \quad u \otimes f \otimes g \otimes f \otimes g \otimes v \quad \Rightarrow \quad u \otimes f \otimes g \otimes v \quad : \quad C \quad \to \quad D$$

which can be represented as
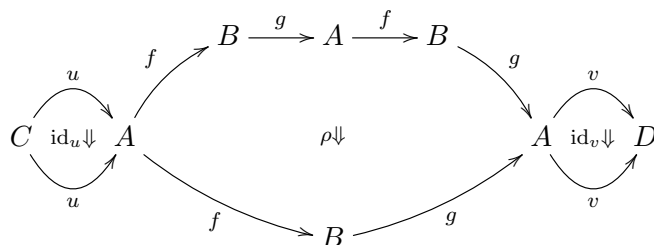
We usually omit drawing the identity 2-cells and simply write
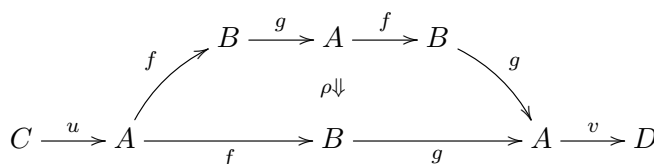
for this 2-cell.

As another example of the fact that 2-cells represent rewriting paths, consider the word $fgfgfg$ which can be rewritten to fg:

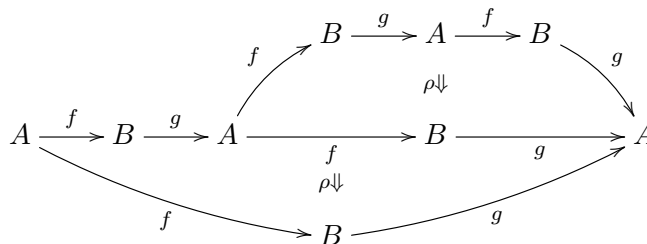$$fgfgfg \quad \Rightarrow \quad fgfg \quad \Rightarrow \quad fg$$

Actually, there are *two* such rewriting paths because, in the first step, the rule ρ can be applied either on the left or on the right of the word. These two distinct paths correspond to the morphisms

$$\rho \circ (\rho \otimes f \otimes g) \qquad \text{and} \qquad \rho \circ (f \otimes g \otimes \rho)$$
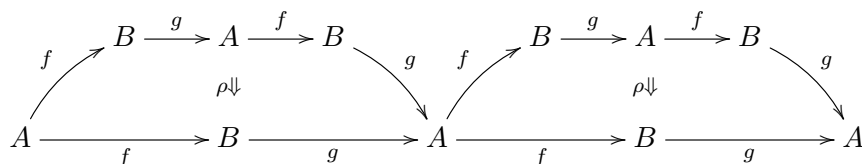
respectively represented diagrammatically by

(1.11)

and

$$
\begin{array}{c}
B \xrightarrow{\;g\;} A \xrightarrow{\;f\;} B \\
\end{array}
$$
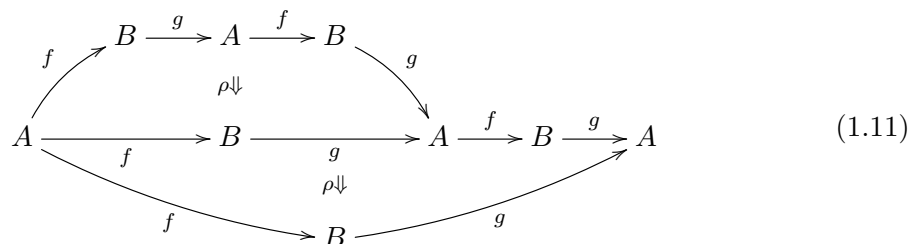
Notice that it is also possible in this setting to express the rewriting of two disjoint parts of a word in parallel, such as

$$
\rho \otimes \rho \quad : \quad fgfgfgfg \quad \Rightarrow \quad fgfg
$$

which is represented diagrammatically by

1.2.6. *String diagrams.* The diagrammatic notation for cells in 2-categories introduced in Section 1.2.4 is quite useful and convenient. However, from such diagrams, it is sometimes difficult to get intuitions about the way morphisms should be manipulated and having another, more "geometrical", representation of morphisms can be very useful. Such a notation was formally introduced by Joyal and Street [17] and is called *string diagrams.*

We shall first explain this notation on an example. Consider the 2-cell ρ in the 2-category described in previous section, whose diagrammatic representation is given in (1.10). We can consider this 2-cell as a device with four typed inputs respectively of type f, g, f and g, which produces two outputs respectively of type f and g. It is thus natural to depict it as
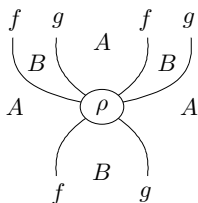
The 2-cell ρ is represented by the point in the center, each of the 1-cells in its input (resp. output) correspond to a wire going in (resp. out of) it. The portions of the plane delimited by the wires correspond to 0-cells at the source or target of the 1-cells corresponding to the delimiting wires. Since, in this representation, 2-cells consist of points linked by strings (or wires), it is called a *string diagram.*

Vertical composition amounts to vertically juxtaposing diagrams and "linking the wires", whereas horizontal composition is given by horizontal juxtaposition of diagrams. The two
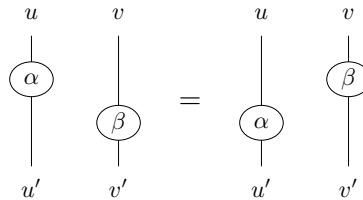
diagrams for composition (1.6) and (1.7) are thus drawn respectively by

$$u$$
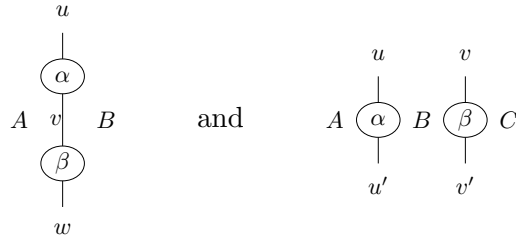
and vertical identities are represented by wires. For example, the string-diagrammatic notation of the morphism (1.11) is
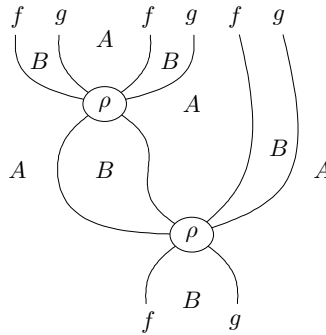
These diagrams should be considered modulo planar continuous deformations. In particular, the Godement exchange law (1.9) is satisfied

because the diagram on the left can be deformed into the diagram on the right, and more generally, it can be checked that all the axioms of 2-categories are satisfied.

1.3. Term and 3-dimensional rewriting systems.

1.3.1. *Term rewriting systems.* In the same way string rewriting systems can be generalized to 2-dimensional rewriting systems, which rewrite rewriting paths in 1-dimensional rewriting systems, term rewriting systems can be generalized to 3-dimensional rewriting systems which rewrite rewriting paths in 2-dimensional rewriting systems.

A *signature* (Σ, a) consists of a set Σ called *alphabet*, whose elements are called *symbols*, together with a function $a : \Sigma \to \mathbb{N}$ which to every symbol $f \in \Sigma$ associates its *arity*. We also suppose fixed a set Var of *variables*. A *term* is either a variable $x \in Var$ or of the form $f(t_1, \ldots, t_n)$ where the t_i are terms and f is a symbol of arity n. Given two terms t and u and a variable x, we denote by $t[u/x]$ the term obtained from t by substituting all

the occurrences of the variable x by u. We write Σ^* for the set of terms generated by the signature Σ. A *substitution* $\sigma : Var \to \Sigma^*$ is a function which to every variable associates a term. Given a term t, we write $t[\sigma]$ for the term obtained from t by replacing every variable x by $\sigma(x)$. A *term rewriting system* R on such a signature is a set $R \subseteq \Sigma^* \times \Sigma^*$ of pairs of terms. A term t *one-step rewrites* to a term u, what we write $t \Rightarrow u$, when there exists a rule $(l, r) \in R$ and a substitution σ such that $u = l[\sigma]$ and $v = r[\sigma]$. We write $\Rightarrow^*$ for the transitive closure of the relation $\Rightarrow$ on terms, and say that a term t *rewrites* to a term u whenever $t \Rightarrow^* u$.

We now generalize the notion of term rewriting system so that it appears as an extension of 2-dimensional rewriting systems. Having done that, we shall be able to see a term as a rewriting path in a 2-dimensional rewriting system: a 3-dimensional rewriting system will thus consist of rewriting rules which relate terms, i.e. rewriting rules which rewrite rewriting paths in a 2-dimensional rewriting system. We should follow the intuition given by string diagrams. For example, if we consider a signature $\Sigma = \{f, g\}$, the two symbols f and g being of arity 2, as mentioned in the introduction, the term $f(x, g(y, z))$, where x, y and z are variables, can be drawn as a tree
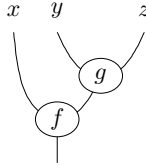
which looks very much like the string-diagrammatic notation for a morphism $f \circ (\mathrm{id} \otimes g)$, whose diagrammatic notation would be
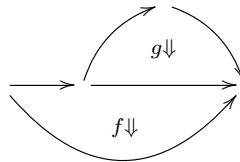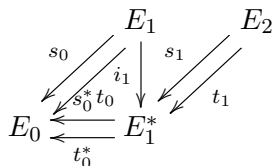

In order to make this intuition formal, recall from Example 1.1 that the monoid $\mathbb{N}$ of integers can be seen as the category generated by the graph with only one vertex and one edge: if we write $E_0 = \{*\}$ for the set of vertices ($*$ being the only vertex) and $E_1 = \{1\}$ for the set of edges (1 being the only edge, with $*$ both as source and target) and follow the notations of Section 1.1.2, we have the diagram
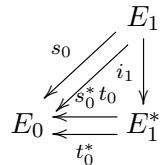
where $E_1^* \cong \mathbb{N}$ is the set of morphisms in the category generated by the graph (i.e. the paths in the graph). For example, the integer 3 corresponds to the path $1 \otimes 1 \otimes 1$:

$$* \xrightarrow{\ 1\ } * \xrightarrow{\ 1\ } * \xrightarrow{\ 1\ } *$$

and more generally an integer n to the composition of n copies of 1. The arity can thus be seen as an arrow $a : \Sigma \to E_1^*$. In order to be consistent with the previous notations, we write E_2 instead of Σ and s_1 instead of a, since the arity denotes the number of inputs. We also write $t_1 : E_2 \to E_2^*$ for the constant function equal to 1: it denotes the *coarity* of a symbol (its number of outputs), which is always 1 in the setting of term rewriting systems. The signature of a term rewriting system can thus be specified by a 2-rewriting system
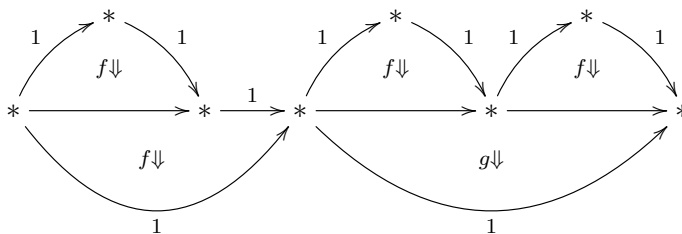
$$
\begin{array}{c}
E_1 \qquad E_2 \\
s_0 \qquad i_1 \quad s_1 \qquad \\
s_0^* \, t_0 \qquad t_1 \\
E_0 \xleftarrow{\quad} E_1^* \\
t_0^*
\end{array}
$$

where $E_0 = \{*\}$, $E_1 = \{1\}$ and t_1 is the constant function equal to 1. For example, the previously considered signature with two symbols f, g of arity 2 is represented by the 2-rewriting system such that

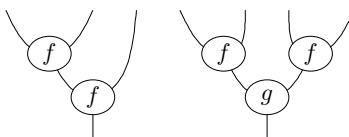$$E_2 = \{f, g\} \quad \text{and} \quad s_1(f) = s_1(g) = 1 \otimes 1 \tag{1.12}$$

Any such signature being a 2-rewriting system, it generates a 2-category $\mathcal{C}$ with E_2^* as set of 2-cells. As previously, we write $i_2 : E_2 \to E_2^*$ for the canonical injection and $s_1^*, t_1^* : E_2^* \to E_1^*$ for the maps associating to each 2-cell its source and target respectively. Since $E_0 = \{*\}$, the 2-category $\mathcal{C}$ has only one 0-cell $*$. Since $E_1 = \{1\}$, the 1-cells of the 2-category $\mathcal{C}$ are in bijection with $\mathbb{N}$ (as explained above). It is easy to show that the 2-cells $\alpha : n \Rightarrow 1 : * \to *$ of $\mathcal{C}$ are in bijection with linear terms with variables in $x_1, \ldots, x_n$: a term is *linear* when all the variables $x_1, \ldots, x_n$ appear exactly once in the term, in this order, e.g. with the signature (1.12), the term $f(x_1, g(x_2, x_3))$ is linear whereas $f(x_1, x_1)$ and $g(x_2, x_1)$ are not. More generally, the 2-cells $\alpha : n \Rightarrow m : * \to *$ are in bijection with linear m-uples of terms with n variables $x_1, \ldots, x_n$. For example, with the signature (1.12), the term

$$(f \circ (f \otimes \mathrm{id}_1)) \otimes (g \circ (f \otimes f))$$

which can be represented diagrammatically as

or string-diagrammatically as

represents the pair $f(f(x_1, x_2), x_3), \; g(f(x_4, x_5), f(x_6, x_7))$.

A term rewriting system operating on linear terms can thus be thought as rewriting rewriting paths in a 2-dimensional rewriting system and be represented as a diagram

$$\text{(1.13)}$$

where E_2 is the alphabet and E_3 is the set of rewriting rules, such that rules rewrite a term into a term of the same type:

$$s_1^* \circ s_2 = s_1^* \circ t_2 \qquad \text{and} \qquad t_1^* \circ s_2 = t_1^* \circ t_2$$

and the three conditions below are satisfied:

(1) the set E_0 is reduced to one element $*$,
(2) the set E_1 is reduced to one element 1,
(3) the function t_2 is the constant function equal to 1.

In the following, we drop those three conditions in the general definition of 3-dimensional rewriting systems. Dropping condition (3) allows us to consider terms with multiple outputs (whereas terms in term-rewriting systems always have exactly one output). In particular, we will see in Section 2.2.2 that this enables us to recover the usual setting of (non-linear) term rewriting systems: non-linearity is obtained by adding to the signature symbols to explicitly duplicate, erase and swap variables. Dropping condition (2) allows us to consider typed terms. For example, a symbol $f \in E_2$ with $s_1(f) = A \otimes B$ and $t_1(f) = C \otimes D$, where $A, B, C, D \in E_1$ means that f has two inputs of respective types A and B and two outputs of respective types C and D; diagrammatically,

Finally, dropping condition (1) allows to have typed types, as explained in Section 1.2.

Notice that usually, term rewriting systems are considered to generalize string rewriting system because a letter can be seen as a symbol of arity 1. While this is true, this is not the right point of view! At the light of the previous explanations, we see that term rewriting systems do generalize string rewriting systems in the sense that they rewrite rewriting paths in string rewriting systems.

1.4. **Polygraphs.** We believe that the general pattern should have become clear now, and one can generalize the situation in order to inductively define a notion of n-dimensional rewriting system. This was formalized by Burroni under the name of n-polygraph [5]. The article introducing polygraphs was in fact rediscovering the notion of n-computad invented 17 years earlier by Street in its 2-dimensional version [36] and later on generalized to higher dimensions by Power [32]. We insist here on the contribution of Burroni, who first found and formalized connections between these general constructions and rewriting theory.

The notion of n-polygraph is defined inductively (on $n \in \mathbb{N}$) as follows. A 0-polygraph is a set E_0. Now suppose given an n-polygraph, i.e. a diagram

$$\tag{1.14}$$

in **Set** together with a structure of $(n-1)$-category on the $(n-1)$-graph

$$E_0^* \underset{t_0^*}{\overset{s_0^*}{\leftleftarrows}} E_1^* \underset{t_1^*}{\overset{s_1^*}{\leftleftarrows}} E_2^* \qquad \cdots \qquad E_{n-2}^* \underset{t_{n-2}^*}{\overset{s_{n-2}^*}{\leftleftarrows}} E_{n-1}^*$$

Such a polygraph generates a free n-category, which will be noted E^*, with the previous $(n-1)$-category as underlying $(n-1)$-category and containing the elements of E_n as n-cells, with source and target indicated by the functions s_{n-1} and t_{n-1}. We write E_n^* for its set of n-cells, $i_n : E_n \to E_n^*$ for the canonical injection and $s_{n-1}^*, t_{n-1}^* : E_n^* \to E_{n-1}^*$ for the source and target maps. An $(n+1)$-polygraph is then defined as a diagram of the form

in **Set**, together with the structure of n-category on the n-graph at the bottom of the diagram, such that

$$s_{n-1}^* \circ s_n = s_{n-1}^* \circ t_n \qquad \text{and} \qquad t_{n-1}^* \circ s_n = t_{n-1}^* \circ t_n$$

We refer the reader to [5] for a detailed definition and to [14] for a more abstract equivalent definition.

Given an n-polygraph (1.14), the elements of E_k are called k-*generators*. Notice that E_0^* is the free 0-category (i.e. set) on E_0, so E_0^* is isomorphic to E_0, and we might as well decide that they are equal: we have implicitly done this assumption in our presentation of n-dimensional rewriting systems in order to ease the explanation. Given two n-polygraphs E and F, a morphism between them consists of a family $(f_k : E_k \to F_k)_{0 \leqslant k \leqslant n}$ of functions such that $s_k \circ f_{k+1} = f_k^* \circ s_k$ for every index k, and we write $\mathbf{Pol}_n$ for the category of n-polygraphs. Every m-polygraph E admits an underlying n-polygraph (with $m > n$), often written $E|_n$, thus inducing a forgetful functor $\mathbf{Pol}_m \to \mathbf{Pol}_n$.

2. Presentations of n-categories

A crucial application of polygraphs relies on the fact that they provide us with a notion of *presentation* of n-categories: an $(n+1)$-polygraph E can be seen as a description of an n-category by generators (the elements of E_k with $0 \leqslant k \leqslant n$ being generators for k-cells) quotiented by relations (the elements of E_{n+1}). More precisely, an $(n+1)$-polygraph E generates an $(n+1)$-category E^*. The n-category *presented* by the polygraph E, written $\overline{E^*}$ is obtained by quotienting the underlying n-category of E^* by the relation identifying two n-cells f and g whenever there exists an $(n+1)$-cell $\alpha : f \to g$ in E^*. More generally, one

says that an n-category $\mathcal{C}$ is *presented* by an $(n+1)$-polygraph E when $\mathcal{C}$ is isomorphic to $\overline{E^*}$.

Presentations of categories are useful in the sense that they can provide us with small (even sometimes finite) descriptions of categories. This idea of presentation generalizes the well-known notion of presentation of a monoid (or a group). We begin by recalling this simple and well studied setting.

2.1. Presentations of categories.

2.1.1. *Presentations of monoids.* One of the main applications of string rewriting systems, that we are going to generalize here, is the construction of presentations of a monoid.

Definition 2.1. A *presentation* $\langle \Sigma \mid R \rangle$ of a monoid M consists of
- a set Σ of *generators*,
- a set $R \subseteq \Sigma^* \times \Sigma^*$ of *relations*

where Σ^* denotes the free monoid over Σ, such that the monoid M is isomorphic to $\Sigma^*/\equiv_R$, where $\equiv_R$ is the smallest congruence (with respect to concatenation of words) on Σ^* containing R.

Example 2.2. The additive monoid $\mathbb{N}$ admits the presentation with one generator a and no relation $\langle a \mid \rangle$. Similarly,

$$\mathbb{N}/2\mathbb{N} \cong \langle a \mid aa = 1 \rangle \qquad \mathbb{N} \times \mathbb{N} \cong \langle a, b \mid ba = ab \rangle$$

and, if we write $\mathfrak{S}_n$ for the monoid of symmetries on n elements,

$$\mathfrak{S}_n \cong \langle \sigma_1, \ldots, \sigma_n \mid \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}, \ \sigma_i^2 = 1, \ \sigma_i \sigma_j = \sigma_j \sigma_i \rangle$$

How does one construct a presentation $\langle \Sigma \mid R \rangle$ of a given monoid M? The main difficulty consists in showing the isomorphism $M \cong (\Sigma^*/\equiv_R)$ which is a priori difficult because the second monoid is described as terms modulo a congruence. One of the great usefulness of string rewriting systems is that, when they are convergent (i.e. terminating and confluent), they provide one with a notion of normal form of terms modulo the congruence generated by the rewriting rules. In order to show the isomorphism $M \cong (\Sigma^*/\equiv_R)$, one can thus try to follow the following recipe:

(1) Orient the relations in R in order to get a string rewriting system on the alphabet Σ.
(2) Show that the rewriting system is terminating.
(3) Compute the critical pairs of the rewriting system and show that they are joinable.
(4) The two previous points ensure that the rewriting system is convergent. Compute the normal forms and show that they are in bijection with the elements of M in a way compatible with multiplication and unit.

It is well known that the joinability of critical pairs checked in (3) implies the local confluence of the rewriting system. Its confluence can then be deduced by Newman's lemma provided that it is terminating (2). The compatibility condition of (4) can be made explicit as follows. We write $(M, \times, 1)$ for the operations of the monoid M, we write $\overline{w}$ for the normal form of a word $w \in \Sigma^*$, and set $\overline{\Sigma^*} = \{\ \overline{w} \mid w \in \Sigma^*\ \}$. One should provide a pair of functions $f : \overline{\Sigma^*} \to M$ and $g : M \to \overline{\Sigma^*}$, which are mutually inverse morphisms of monoids:

$$f(\overline{\overline{u} \otimes \overline{v}}) = f(\overline{u}) \times f(\overline{v}) \qquad f(\overline{\varepsilon}) = 1 \qquad g(m \times n) = g(m) \otimes g(n) \qquad g(1) = \overline{\varepsilon}$$

for every $u, v \in \Sigma^*$ and $m, n \in M$, where $\otimes$ denotes concatenation and ε the empty word.

Example 2.3. We can show that the additive monoid $M = \mathbb{N} \times (\mathbb{N}/2\mathbb{N})$ admits the presentation $\langle a, b \mid ba = ab, 1 = bb \rangle$, where 1 denotes the empty word:

(1) We orient the rules as follows:

$$ba \Rightarrow ab \qquad \text{and} \qquad bb \Rightarrow 1$$

(2) The system is terminating: introduce a suitable measure on words based on the fact that the rules respectively decrease the number of a after a b, and the total number of b in a word.

(3) The two critical pairs are joinable:
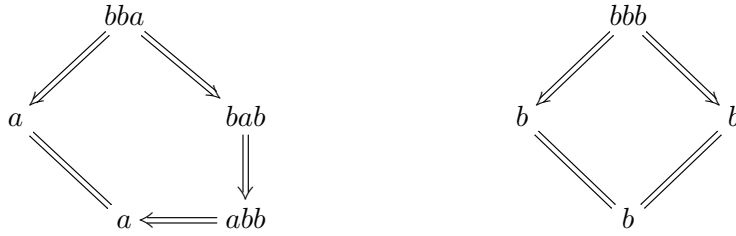
(4) The normal forms are words of the form $a^n b^m$ with $n \in \mathbb{N}$ and $m \in \{0, 1\}$. These are obviously in bijection with elements of $\mathbb{N} \times (\mathbb{N}/2\mathbb{N})$ and the bijection can be shown to be a morphism of monoids (see below).

Since normal forms are canonical representatives of elements of the free monoid Σ^* over the alphabet Σ modulo the congruence $\equiv_R$, in order to define the morphism f witnessing the bijection (4), by universal property of the free monoid it is enough to define f on letters in Σ, extend it as a morphism $f : \Sigma^* \to M$ and show that the images of two words equivalent with respect to $\equiv_R$ are equal, which can be checked by showing that the image by f of any left member of any rule is equal the corresponding right member. Of course, the explicit construction of g is not necessary and it is equivalent (and sometimes more natural) to show that f is both surjective and injective.

Example 2.4. The last step of Example 2.3 can thus be shown as follows:

(4) We define a morphism $f : \overline{\Sigma^*} \to M$ as follows. The morphism f is defined on letters by $f(a) = (1, 0)$, $f(b) = (0, 1)$. The induced morphism $f : \Sigma^* \to M$ is compatible with the congruence $\equiv_R$:

$$f(ba) = f(b) + f(a) = (0, 1) + (1, 0) = (1, 1) = (1, 0) + (0, 1) = f(a) + f(b) = f(ab)$$

and

$$f(bb) = f(b) + f(b) = (0, 1) + (0, 1) = (0, 0) = f(1)$$

Conversely, g is defined by $g((m, n)) = a^m b^n$, which is a morphism of monoids, i.e.

$$g((m, n) + (m', n')) = a^{m+m'} b^{n+n'} = \overline{a^m b^n a^{m'} b^{n'}}$$

and the two morphisms are mutually inverse

$$g \circ f(a^m b^n) = g((m, n)) = a^m b^n \quad \text{and} \quad f \circ g((m, n)) = f(a^m b^n) = (m, n)$$

This general methodology can be straightforwardly adapted to n-polygraphs in order to build presentations of n-categories. We detail a few interesting particular cases in the following.

Remark 2.5. Notice that the method described above is only a "recipe" for constructing presentations, which works (or can be adapted) in many cases. In fact, it has been shown by Squier [35, 23], using homological methods, that there exists monoids which admit finite presentations, but no finite presentation which can be oriented into a convergent rewriting system. These theoretical considerations aside, convergent presentations are sometimes difficult to build: in these cases, the following weaker method (used for example in [22]) can be tried:

(2) Introduce a notion of canonical form for words (which will play the role of previous normal forms).
(3) Show that every word reduces to a canonical form.
(4) Define a morphism $f : \Sigma^* \to M$ which is compatible with rewriting rules as previously, and show that it induces a bijection between the canonical forms and the elements of M.

This variant has the advantage of not requiring to show termination, but it can lead one to have to consider many cases (many more than critical pairs when the system is convergent).

2.1.2. *Presentations of categories.* 2-polygraphs more generally present categories: the case of a presentation of a monoid is the particular case where the presented category has only one object. Namely, a 2-polygraph E describes a category which is the smallest category with the 0-generators in E_0 as objects, containing the typed 1-generators in E_1 as morphisms, quotiented by the relations in E_2.

Example 2.6. Consider the simplicial category Δ: its objects are natural integers $n \in \mathbb{N}$, where n is seen as the totally ordered set with n elements $\{0, 1, 2, \ldots, n-1\}$ and morphisms $f : m \to n$ are increasing functions. This category is presented by the 2-polygraph E with $E_0 = \mathbb{N}$, the set E_1 contains the two families of 1-generators indexed by integers $n, i \in \mathbb{N}$, such that $0 \leqslant i \leqslant n$,

$$\mu_i^{n+1} : n + 2 \to n + 1 \qquad \text{and} \qquad \eta_i^n : n \to n + 1$$

and the set E_2 of rewriting rules contains the families of 2-generators index by $n, i \in \mathbb{N}$

$$\begin{aligned}
\mu_j^{n+1}\mu_i^{n+2} &\Rightarrow \mu_i^{n+1}\mu_{j+1}^{n+2} \quad \text{for } i \leqslant j, \\
\eta_i^{n+1}\eta_j^n &\Rightarrow \eta_{j+1}^{n+1}\eta_i^n \quad \text{for } i \leqslant j, \\
\mu_j^{n+1}\eta_i^{n+1} &\Rightarrow \begin{cases} \eta_i^n \mu_{j-1}^n & \text{for } i < j, \\ \mathrm{id}_{n+1} & \text{for } i = j \text{ or } i = j+1, \\ \eta_{i-1}^n \mu_j^n & \text{for } i > j+1 \end{cases}
\end{aligned}$$

In order to show this, the same recipe as previously can be used: the rewriting system can be shown to be terminating and confluent, the normal forms being terms of the form

$$\eta_{i_k}^{m-h+k-1} \circ \ldots \circ \eta_{i_1}^{m-h} \circ \mu_{j_h}^{m-h} \circ \ldots \circ \mu_{j_1}^{m-1} \quad : \quad m \to n$$

such that $n = m - h + k$, $n > i_k > \ldots > i_1 \geq 0$ and $0 \leqslant j_h < \ldots < j_1 < m$. We can then construct a functor $F : \overline{E^*} \to \Delta$, which is defined as the identity on objects and is defined on 1-generators by

$$F(\mu_i^n) = k \mapsto \begin{cases} k & \text{if } k \leqslant i \\ k-1 & \text{if } k > i \end{cases} \quad \text{and} \quad F(\eta_i^n) = k \mapsto \begin{cases} k & \text{if } k < i \\ k+1 & \text{if } k \geq i \end{cases}$$

The image of the left member of a rule by F can be checked to be equal to the image of the corresponding right member and it can be shown to be a bijection, see [26] (section VII.5) for details. A refined way to construct this presentation is hinted in Example 2.18.

2.2. **Presentations of 2-categories.**

2.2.1. *Monoidal categories.* Before explaining how term rewriting systems provide us with a notion of presentation of a Lawvere theory, we need to introduce the notion of monoidal category. Formally, a monoidal category could be defined as a 2-category (Definition 1.3) with exactly one 0-cell. However, in the same way that a category with only one object can be reformulated as a monoid (i.e. a set with operations), a monoidal category is generally defined as a category with operations.

Definition 2.7. A *strict monoidal category* $(\mathcal{C}, \otimes, I)$ consists of a category $\mathcal{C}$ together with
  – a functor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$, called *tensor product*,
  – an object I of $\mathcal{C}$, called *unit*,
such that
  – tensor product is associative: for every objects A,B and C, $(A \otimes B) \otimes C = A \otimes (B \otimes C)$
  – units are neutral elements: for every object A, $I \otimes A = A = A \otimes I$.

The tensor product $f \otimes g$ of two morphisms f and g should be thought as the morphism f "in parallel" with the morphism g. This definition is a particular case of a more general definition of monoidal categories [26] (which is why it is called *strict*), however we will only consider this variant here. A monoidal category $(\mathcal{C}, \otimes, I)$ as above gives rise to a 2-category, with only one 0-cell, the 1-cells being the objects of $\mathcal{C}$, the 2-cells being the morphisms of $\mathcal{C}$, vertical composition being given by the composition of $\mathcal{C}$ and horizontal composition being given by $\otimes$; and this induces an isomorphism between the category of monoidal categories and the categories of 2-categories with one object (this is a generalization of the situation between monoids and categories described in Section 1.2.2). In other words, a monoidal category is "the same as" a 2-category with one object and we can in particular reuse the string-diagrammatic notation for morphisms. We thus sometimes implicitly consider a monoidal category as a 2-category in the following.

Any cartesian category $\mathcal{C}$ can be equipped with a structure of monoidal category $(\mathcal{C}, \times, 1)$, tensor product being cartesian product and unit being the terminal object – for simplicity, we consider that cartesian products are strictly associative, which is always true up to equivalence of categories, but this assumption could have been dropped if we had worked with the more general notion of (non-strict) monoidal category. The usual notion of monoid can be generalized from the monoidal category $(\mathbf{Set}, \times, 1)$ to any monoidal category as follows.

Definition 2.8. A *monoid* (M, μ, η) in a monoidal category $(\mathcal{C}, \otimes, I)$ consists of an object M equipped with two morphisms

$$\mu : M \otimes M \to M \quad \text{and} \quad \eta : I \to M$$

such that the diagrams

$$
\begin{array}{ccc}
M \otimes M \otimes M & \xrightarrow{\mu \otimes \mathrm{id}_M} & M \otimes M \\
{\scriptstyle \mathrm{id}_M \otimes M} \downarrow & & \downarrow {\scriptstyle \mu} \\
M \otimes M & \xrightarrow{\mu} & M
\end{array}
\qquad \text{and} \qquad
\begin{array}{ccc}
I \otimes M \xrightarrow{\eta \otimes \mathrm{id}_M} & M \otimes M & \xleftarrow{\mathrm{id}_M \otimes \eta} M \otimes I \\
{\scriptstyle \mathrm{id}_M} \searrow & \downarrow {\scriptstyle \mu} & \swarrow {\scriptstyle \mathrm{id}_M} \\
& M &
\end{array}
$$

commute. String-diagrammatically,

Example 2.9. A monoid in $(\mathbf{Set}, \times, 1)$ is a monoid in the usual sense, a monoid in $(\mathbf{Cat}, \times, 1)$ is a monoidal category.

As mentioned before, the tensor product intuitively expresses the possibility of having two morphisms done in parallel. One sometimes also needs to be able to "swap" two morphisms, which is formalized by the notion of symmetric monoidal category.

Definition 2.10. A *symmetric monoidal category* $(\mathcal{C}, \otimes, I, \gamma)$ consists of a monoidal category $(\mathcal{C}, \otimes, I)$ together with a family γ of isomorphisms $\gamma_{A,B} : A \otimes B \to B \otimes A$, indexed by pairs of objects A, B of $\mathcal{C}$ and called *symmetry*, which is natural in the sense that for every morphisms $f : A \to A'$ and $g : B \to B'$,

$$
\begin{array}{ccc}
A \otimes B & \xrightarrow{f \otimes g} & A' \otimes B' \\
{\scriptstyle \gamma_{A,B}} \downarrow & & \downarrow {\scriptstyle \gamma_{A',B'}} \\
B \otimes A & \xrightarrow[g \otimes f]{} & B' \otimes A'
\end{array}
$$

and such that for every objects A, B and C the diagrams

commute, and for every objects A and B,

$$\gamma_{B,A} \circ \gamma_{A,B} \quad = \quad \mathrm{id}_{A \otimes B}$$

Again, every cartesian category can be equipped with a symmetry using the canonical isomorphisms $A \otimes B \cong B \otimes A$ induced by the cartesian product. The notion of commutative monoid can thus be generalized to any symmetric monoidal category as follows.

Definition 2.11. A *commutative monoid* (M, μ, η) in a symmetric monoidal category $(\mathcal{C}, \otimes, I, \gamma)$ is a monoid such that

$$\mu \circ \gamma_{M,M} \quad = \quad \mu$$

String-diagrammatically,

$$
\begin{array}{ccc}
M \quad M & & M \quad M \\
\gamma_{M,M} & = & \\
\mu & & \mu \\
M & & M
\end{array}
$$
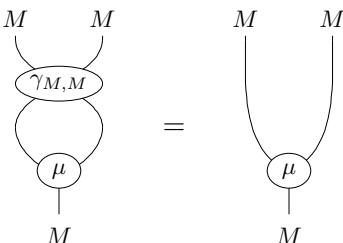
The notion of *commutative comonoid* (M, δ, ε) with $\delta : M \to M \otimes M$ and $\varepsilon : M \to I$ is defined dually. Interestingly, this notion allows us to characterize cartesian categories among monoidal categories:

Proposition 2.12. *A monoidal category* $(\mathcal{C}, \otimes, I)$ *is cartesian, with* $\otimes$ *as cartesian product and* I *as terminal object, if and only if the category can be equipped with a symmetry* γ *and every object* A *can be equipped with a structure of commutative comonoid* $(A, \delta_A, \varepsilon_A)$ *which is natural in the sense that for every morphism* $f : A \to B$ *the diagrams*

$$
\begin{array}{ccc}
A & \xrightarrow{f} & B \\
\delta_A \downarrow & & \downarrow \delta_B \\
A \otimes A & \xrightarrow{f \otimes f} & B \otimes B
\end{array}
\qquad and \qquad
\begin{array}{ccc}
A & \xrightarrow{\quad f \quad} & B \\
& \varepsilon_A \searrow \quad \swarrow \varepsilon_B & \\
& I &
\end{array}
$$

commute; string-diagrammatically,
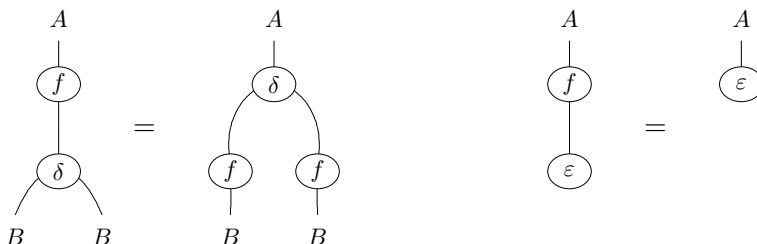
This result was considered for a long time as folklore in category theory. A detailed proof of this fact is not really difficult and can for example be found in [27]. The intuition behind this result is however very enlightening: a cartesian category is a monoidal category in which any object A can be duplicated (by $\delta_A : A \to A \otimes A$), erased (by $\varepsilon_A : A \to I$) or swapped with another object B (by $\gamma_{A,B} : A \otimes B \to B \otimes A$)! These operations can also be thought as the analogous of contraction, weakening and exchange rules respectively in sequent calculus. We use this result below in order to show that in fact presentations of Lawvere theories are a particular case of presentations of 2-categories.

2.2.2. *Presentations of Lawvere theories.* We have seen that string rewriting systems correspond to presentations of monoids. Similarly, we may wonder: what are term rewriting systems presentations of? It turns out that the right answer for this are particular categories, called Lawvere theories, which were introduced by Lawvere in his thesis [24].

Definition 2.13. A *Lawvere theory* is a cartesian category whose objects are integers $\mathbb{N}$ such that 0 is a terminal object and product is given on objects by addition.

Example 2.14. A Lawvere theory **Mat** of $\mathbb{N}$-valued matrices can be defined as follows. Its objects are natural numbers and a morphism $M : m \to n$ is a matrix M of size $n \times m$ (with n rows and m columns) with coefficients in $\mathbb{N}$. The composition $N \circ M : m \to p$ of two morphisms $M : m \to n$ and $N : n \to p$ is given by the usual composition $N \circ M = NM$ of matrices and identities are usual identity matrices. This category is cartesian with cartesian product defined on objects by addition and the cartesian product of two morphisms $M : m \to m'$ and $N : n \to n'$ is their direct sum, i.e. the matrix $M \times N : m + n \to m' + n'$ defined as

$$\begin{pmatrix} M & 0 \\ 0 & N \end{pmatrix}$$

using a block representation of the matrix.

The link with term rewriting systems can be explained as follows. Suppose given a signature (Σ, a). This signature induces a Lawvere theory $\mathcal{L}_\Sigma$ whose morphisms $t : m \to n$ are n-uples $(t_1, \ldots, t_n)$ of terms using variables in $X_m = \{x_1, \ldots, x_m\}$. The composition $u \circ t : m \to p$ of two morphisms $t : m \to n$ and $u : n \to p$ is the morphism v defined by substitution as

$$(u_1[t_1/x_1, \ldots, t_n/x_n] \, , \, \ldots \, , \, u_p[t_1/x_1, \ldots, t_n/x_n])$$

and the identity on an object n is $(x_1, \ldots, x_n)$. This category is cartesian, in particular the structure of commutative comonoid $(1, \delta : 1 \to 2, \varepsilon : 1 \to 0)$ on the object 1 given by Proposition 2.12 is defined by $\delta = (x_1, x_1)$, $\varepsilon = ()$ and the symmetry on 1 is $\gamma_{1,1} = (x_2, x_1)$.

Suppose given a term rewriting system R on Σ. We write $\equiv_R$ for the smallest congruence on terms containing the rules in R. A term rewriting system R *presents* a Lawvere theory $\mathcal{C}$ when $\mathcal{C}$ is isomorphic to $\mathcal{L}_\Sigma / \equiv_R$ (the Lawvere theory generated by the signature Σ whose morphisms are quotiented by the equivalence relation $\equiv_R$). Notice that, in order to show the isomorphism, one can try to apply the method given in Section 2.1.1.

Example 2.15. Consider a term rewriting system corresponding to commutative monoids: the signature contains two symbols m of arity 2 and e of arity 0, and there are four rewriting rules

$$m(m(x,y),z) \Rightarrow m(x,m(y,z)) \qquad m(e,x) \Rightarrow x \qquad m(x,e) \Rightarrow x \qquad m(x,y) \Rightarrow m(y,x)$$

This rewriting system presents the Lawvere theory **Mat** of $\mathbb{N}$-valued matrices described in Example 2.14. In order to show this, one could try to show that the rewriting system is convergent, but this fails immediately: the rewriting system is not terminating because of the last rule for commutativity. It is however possible to overcome this difficulty by working on terms modulo commutativity, but we do not detail this here since a more satisfactory way of proving the result is mentioned in Example 2.17. In order to provide an intuition about the result we shall however describe the functor $F : (\mathcal{L}_\Sigma / \equiv_R) \to$ **Mat** witnessing the isomorphism between the two categories, which can be shown to preserve cartesian products. The functor F is the identity on objects. Suppose given a morphism $t : m \to n$, constituted of an n-uple of terms $(t_1, \ldots, t_n)$ with variables in X_m. The image $F(t)$ of the morphism t is the matrix M of size $n \times m$ such that the entry $M_{i,j}$ is the number of occurrences of the variable x_j in the term t_i. For example, the image of the morphism

$$m(m(x_1, x_1), x_2), \ e, \ x_2 \quad : \quad 2 \to 3$$

is the matrix

$$\begin{pmatrix} 2 & 1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

The functor F can be shown to be compatible with the rewriting rules, i.e. the images of two morphisms equivalent modulo $\equiv_R$ are equal, for example the morphism

$$m(x_1, m(x_2, x_1)), \; m(e, e), \; m(e, x_2) \quad : \quad 2 \to 3$$

is equivalent to the morphism above and has the same image under F. Finally, the functor F can be shown to be an isomorphism.

The result above is quite interesting because it provides one with a concrete description of the category generated by the rewriting system which describes the notion of commutative monoid in a cartesian category: from the result in the example above, it is easy to deduce the somewhat surprising fact that the category **Mat** "impersonates" the notion of monoid in the sense that a monoid in a cartesian category $\mathcal{C}$ is "the same as" a functor **Mat** $\to \mathcal{C}$ which preserves cartesian products (more formally, the category of product-preserving functors from **Mat** to $\mathcal{C}$ and cartesian natural transformations between them is equivalent to the category of monoids in $\mathcal{C}$ and morphisms of monoids).

We have explained in Section 1.3.1 that a 3-polygraph (1.13) with $E_0 = \{*\}$ and $E_1 = \{1\}$ reduced to exactly one element and t_1 indicating that all the 2-generators have exactly one output reformulate linear term rewriting systems. At the light of Proposition 2.12, if we allow 2-generators with multiple outputs, (non-linear) term rewriting systems can be represented by adding 2-generators

$$\delta : 1 \Rightarrow 1 \otimes 1 \qquad \varepsilon : 1 \Rightarrow 0 \qquad \gamma : 1 \otimes 1 \Rightarrow 1 \otimes 1$$

which will allow explicit duplication, erasure and swapping of variables. This was first formalized in [5]:

Proposition 2.16. *Suppose given a term rewriting system R on a signature (Σ, a). The Lawvere theory $\mathcal{L}_\Sigma / \equiv_R$ generated by the rewriting system (seen as a monoidal category and thus as a 2-category) is isomorphic to the 2-category $\overline{E^*}$ presented by the 3-polygraph E with*

$$E_0 = \{*\} \qquad E_1 = \{1\} \qquad E_2 = \Sigma \uplus \{\delta, \varepsilon, \gamma\} \qquad E_3 = R \uplus C$$

with the expected source and target maps, where the rules in C express the fact that $(1, \delta, \varepsilon)$ has a natural structure of commutative comonoid with γ generating the symmetry.

We shall make the source and target maps and the rules in C a bit more explicit. If we write n, as previously, for the tensor of n copies of the object 1, the map $s_1 : E_2 \to E_1^*$ is defined by

$$s_1(f) = a(f) \qquad s_1(\delta) = 1 \qquad s_1(\varepsilon) = 1 \qquad s_1(\gamma) = 2$$

for any symbol $f \in \Sigma$ of arity $a(f)$, and similarly t_1 is defined by

$$t_1(f) = 1 \qquad t_1(\delta) = 2 \qquad t_1(\varepsilon) = 0 \qquad t_1(\gamma) = 2$$

The rules in C express that $(1, \delta, \varepsilon)$ should be a commutative comonoid:

$$(\delta \otimes \mathrm{id}_1) \circ \delta \Rightarrow (\mathrm{id}_1 \otimes \delta) \circ \delta \qquad (\varepsilon \otimes \mathrm{id}_1) \circ \delta \Rightarrow \mathrm{id}_1 \qquad (\mathrm{id}_1 \otimes \varepsilon) \circ \delta \Rightarrow \mathrm{id}_1 \qquad \gamma \circ \delta \Rightarrow \delta$$

and the morphisms corresponding to generators should be natural with respect to every other morphism corresponding to a generator, e.g. for every element $f : 1 \Rightarrow 1$ of E_2, C should also contain the rules

$$\delta \circ f \Rightarrow (f \otimes f) \circ \delta \qquad \varepsilon \circ f \Rightarrow \varepsilon \qquad \gamma \circ (f \otimes \mathrm{id}_1) \Rightarrow (\mathrm{id}_1 \otimes f) \circ \gamma \qquad \gamma \circ (\mathrm{id}_1 \otimes f) \Rightarrow (f \otimes \mathrm{id}_1) \circ \gamma$$

(similar rules should also be introduced for 2-generators f whose arity or coarity is different from 1, see [5]). Finally, the rules in R have source and target the terms corresponding to their left and right member "translated" to explicit manipulations of variables (by using the morphisms δ, ε and γ) followed by a linear term. For example, if Σ contains two symbols f and g of arity 2, a rule $f(g(x_2, x_1), x_2) \Rightarrow g(x_1, x_1)$ will be translated as

$$f \circ (g \otimes \mathrm{id}_1) \circ (\gamma \otimes \mathrm{id}_1) \circ (\mathrm{id}_1 \otimes \delta) \quad \Rightarrow \quad g \circ (\delta \otimes \varepsilon)$$

which is maybe best understood with the corresponding string diagrammatic notation
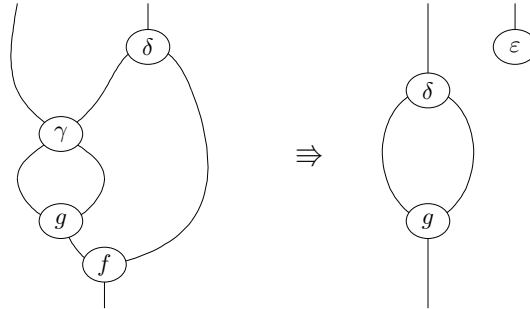
(which is again very close to the usual notation of sharing graphs). This way of representing variables in terms has been the starting point for numerous series of works trying to formalize and axiomatize variable binding, e.g. [6].

Example 2.17. The 3-polygraph corresponding to the term rewriting system of monoids introduced in Example 2.15 is the theory of bicommutative bialgebras, and this polygraph can be shown directly to present the 2-category **Mat** using rewriting techniques [22, 28, 29]. A particularly interesting point is that since swapping of variables is now explicit, the rule expressing commutativity of the monoid is now expressed as
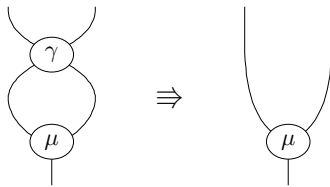
where γ and μ are the 2-generators corresponding to symmetry and multiplication respectively: the rule is not anymore a priori an obstacle to termination (contrarily to the approach of Example 2.15) because it makes the number of γ generators decrease in the morphisms. However, no convergent rewriting system for bicommutative bialgebras is currently known.

2.2.3. *Presentations of 2-categories.* 3-polygraphs provide us with a general notion of presentation by 0-, 1- and 2-generators and relations (the 3-generators) of a 2-category. Many examples of such presentations have been studied by Lafont [22]. We briefly recall here some

fundamental examples that he discovered. The notion of critical pair of the corresponding rewriting systems will be formally introduced in Section 3.2.

Example 2.18. In Example 2.6, we have recalled the presentation of the simplicial category Δ. An unsatisfactory point about this presentation is that it is infinite (it has an infinite number of generators and of relations). Interestingly, one can provide a finite presentation of this category by considering it as a 2-category (more precisely as a monoidal category). Namely, a tensor product $\otimes$ on the category can be defined on objects m and n by $m \otimes n = m + n$ and on morphisms $f : m \to m'$ and $g : n \to n'$ as the morphism $f \otimes g : m + n \to m' + n'$

$$
f \otimes g \quad = \quad i \mapsto \begin{cases} f(i) & \text{if } i < m \\ g(i - m) + m' & \text{if } i \geq m \end{cases}
$$

This induces a structure of monoidal category on Δ, with 0 as unit. This monoidal category admits a presentation by the 3-polygraph E corresponding to the theory of monoids (see Definition 2.8), with $E_0 = \{*\}$, $E_1 = \{1\}$, $E_2 = \{\mu : 2 \to 1, \eta : 0 \to 1\}$ and the 3-generators being

$$a : \mu \circ (\mu \otimes \mathrm{id}_1) \Rrightarrow \mu \circ (\mathrm{id}_1 \otimes \mu) \qquad l : \mu \circ (\eta \otimes \mathrm{id}_1) \Rrightarrow \mathrm{id}_1 \qquad r : \mu \circ (\mathrm{id}_1 \otimes \eta) \Rrightarrow \mathrm{id}_1$$
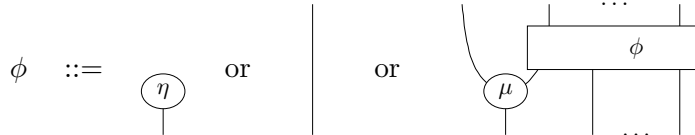
with the following string diagrammatic representation
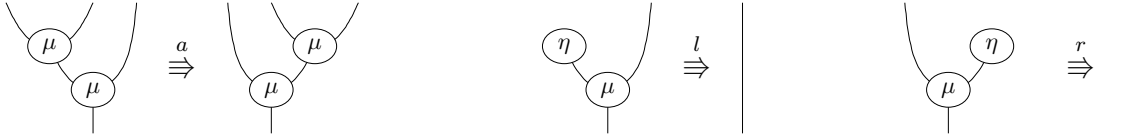
This rewriting system is terminating and its five critical pairs
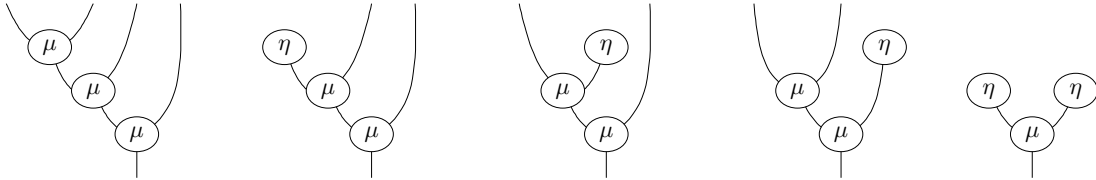
are joinable (see Section 3.1 for a precise definition of the critical pairs in a 3-polygraph). The normal forms are described by the following grammar

One can define a functor $F : \overline{E^*} \to \Delta$ as follows. The functor is the identity on objects. The image of $\mu : 2 \to 1$ is the constant function $m : 2 \to 1$ (which to 0 and 1 associates 1) and the image of $\eta : 0 \to 1$ is the constant function $e : 0 \to 1$. Finally, the functor F can be shown to be an isomorphism (it is easy to show that there is a one-to-one correspondence between normal forms in E and functions in Δ). The fact that F is full means that every function can be expressed as a composite and tensor of m and e. For example, the function $f : 4 \to 3$

whose graph is pictured on the left (such that $f(0) = 1$, $f(1) = 1$, $f(2) = 1$ and $f(3) = 2$) is the image of the morphism on the right:
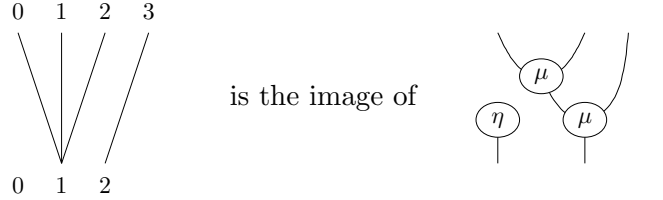

Of course there are many ways to express a given function f using m and e. The fact that the functor is faithful expresses the fact that if two morphisms obtained by composing and tensoring μ and η have the same image by F, then they are equivalent modulo the rewriting rules. Notice that the result of Example 2.6 can be recovered by defining $\mu_i^{n+1} = \mathrm{id}_i \otimes \mu \otimes \mathrm{id}_{n-i}$ and $\eta_i^n = \mathrm{id}_i \otimes \eta \otimes \mathrm{id}_{n-i}$.

Example 2.19. The monoidal category **Bij** is defined similarly as the simplicial category Δ excepting that its morphisms are bijective functions. This monoidal category admits a presentation by the 3-polygraph E corresponding to the theory of symmetries, with $E_0 = \{*\}$, $E_1 = \{1\}$, $E_2 = \{\gamma : 2 \to 2\}$ and the 3-generators being

$$(\gamma \otimes \mathrm{id}_1) \circ (\mathrm{id}_1 \otimes \gamma) \circ (\gamma \otimes \mathrm{id}_1) \Rightarrow (\mathrm{id}_1 \otimes \gamma) \circ (\gamma \otimes \mathrm{id}_1) \circ (\mathrm{id}_1 \otimes \gamma) \qquad \gamma \circ \gamma \Rightarrow \mathrm{id}_2$$

whose string diagrammatic representation is
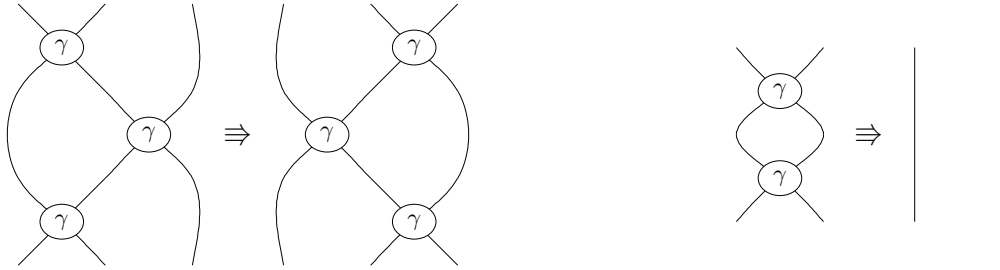
The monoid of endomorphisms of an object n in the category **Bij** is the symmetric group $\mathfrak{S}_n$ (seen as a monoid). In this sense, the polygraph above provides a finite presentation of all the symmetric groups at once.

Interestingly, the preceding rewriting system can be shown to be convergent even though it has an infinite number of critical pairs. Namely, it has the three following obvious critical

pairs

$$(2.1)$$
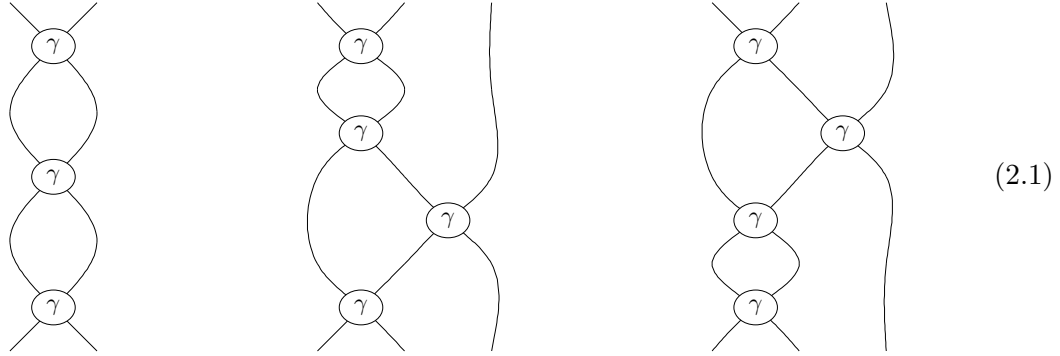
Moreover, for every morphism $\phi : (1 + m) \to (1 + n)$, the morphism (2.2)


$$(2.2)$$
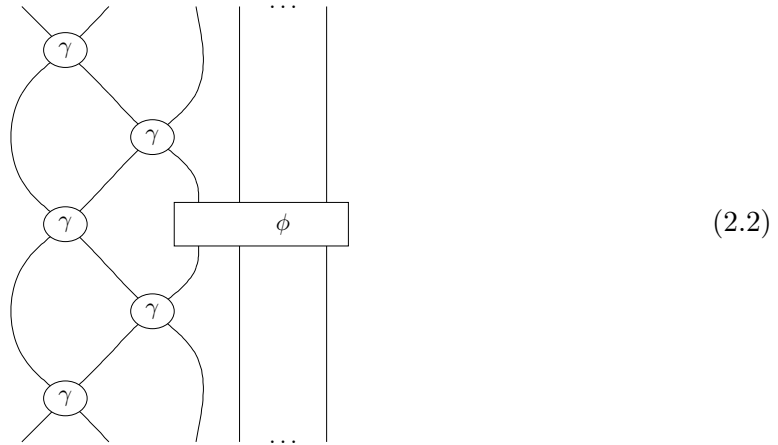
can be rewritten in two different ways, giving rise to an infinite number of critical pairs. Yet, the rewriting system can be shown to be convergent [22]. This contrasts with string or term rewriting systems, which always admit a finite number of critical pairs when they are finite. The difference here seems to come essentially from the fact that the generator γ has multiple outputs as well as multiple inputs.

More recently, these tools have also been applied to more unexpected fields of computer science. For example, the author has proposed a 3-polygraph presenting a monoidal category of game semantics for first order propositional logic [29].

3. REPRESENTING 2-DIMENSIONAL CRITICAL PAIRS

One of the main achievement of rewriting theory is to provide us with algorithms to compute the critical pairs [1], which are at the basis of many advanced tools, to automatically check the confluence of rewriting systems or do Knuth-Bendix completions for example. The fact that the number of critical pairs might be infinite for a finite rewriting system seems to indicate that there is little hope to extend these nice techniques to higher dimensions. We introduce here new theoretical tools in order to overcome this difficulty.

In 3-dimensional rewriting systems, it turns out that we can nevertheless recover a finite description of the critical pairs if we allow ourselves to consider a more general notion of critical pair. For example, in the case of the presentation of the category **Bij** given in Example 2.19, the rewriting system admits a finite number of "critical pairs" if we consider the diagram on the left of (3.1) as a morphism:

$$\tag{3.1}$$
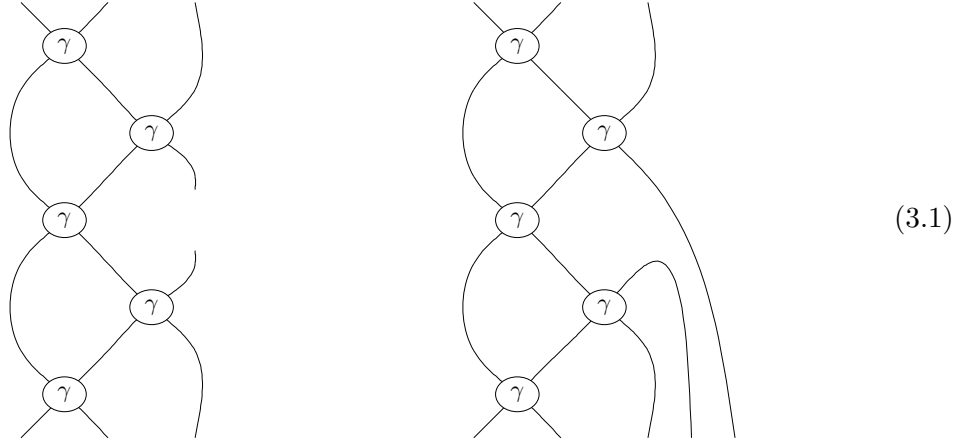
Of course, this "diagram" does not formally make sense: it is not a proper string diagram in the usual sense [17], because of the "punched hole" in the right border. However, one can make this intuitive approach precise by embedding the 2-category of terms into the free 2-category with adjoints it generates, which string diagrammatically corresponds to adding the possibility of "bending wires". The diagram on the left of (3.1) will thus be actually formalized by a diagram such as the one on the right.

This section certainly constitutes the most novel part of the present paper. However, the purpose of this article was to introduce the reader to the concept of higher-dimensional rewriting theory, and to motivate the further developments which are described below. We detail the construction of the multicategory of compact contexts generated by a 2-polygraph E and show that the 2-category E^* it generates can be embedded into it. We also describe how this setting can be used in order to formulate a unification algorithm for 3-polygraphs. A preliminary version of a formal exposition of these later works can be found in [31].

3.1. **The multicategory of contexts.** In order to formalize the notion of critical pair for a 3-polygraph, we need to first formalize the notion of context in the category E^* generated by a 2-polygraph E. This methodology can easily be generalized to n-polygraphs, but we only describe it here in the case of dimension 3 for clarity. Intuitively, a context is a morphism with multiple typed "holes" or "metavariables". Since such a context can have multiple inputs (i.e. multiple holes) and one output (the morphism resulting from filling the holes with morphisms), the contexts are naturally structured as a multicategory [25].

Definition 3.1. A *multicategory* $\mathcal{M}$ (or *colored operad*) is given by
- a class $\mathcal{M}_0$ of *objects*,
- a class $\mathcal{M}_1(A_1, \ldots, A_n; A)$ of *n-ary operations* for every objects $A_1, \ldots, A_n$ and A, we write $f : A_1, \ldots, A_n \to A$ to indicate that $f \in \mathcal{M}_1(A_1, \ldots, A_n; A)$,

– a *composition* function which to every family of operations $f_i : A_i^1, \ldots, A_i^{k_i} \to A_i$, with $1 \leqslant i \leqslant n$, and $f : A_1, \ldots, A_n \to A$, associates a composite operation

$$f \circ (f_1, \ldots, f_n) \quad : \quad A_1^1, \ldots, A_1^{k_1}, \ldots, A_n^1, \ldots, A_n^{k_n} \to A$$

that we often simply write $f(f_1, \ldots, f_n)$,
– an operation $\mathrm{id}_A : A \to A$, called *identity*, for every object A,

such that
– composition is associative:

$$f \circ \Big(f_1 \circ (f_1^1, \ldots, f_1^{k_1}), \ldots, f_n \circ (f_n^1, \ldots, f_n^{k_n}) \Big)$$

$$= \quad (f \circ (f_1, \ldots, f_n)) \circ (f_1, \ldots, f_1^{k_1}, \ldots, f_n^1, \ldots, f_n^{k_n})$$

for every choice of operations f, f_i and f_i^j for which the compositions make sense,
– identities are neutral elements for composition: for every $f : A_1, \ldots, A_n \to A$, we have $f \circ (\mathrm{id}_A, \ldots, \mathrm{id}_A) = f$.

A *symmetric multicategory* is a multicategory $\mathcal{M}$ together with a bijection between the classes $\mathcal{M}(A_1, \ldots, A_n; A)$ and $\mathcal{M}(A_{\sigma(1)}, \ldots, A_{\sigma(n)}; A)$ of operations, for every permutation $\sigma : n \to n$, these bijections having to satisfy some coherence axioms that will be omitted here.

Suppose given a 2-polygraph E. Given a pair of 0-cells $A, B \in E_0^*$ and a pair of parallel 1-cells $f, g : A \to B$ in E_1^*, we write $E[f \Rightarrow g]$ for the polygraph obtained from E by adding a new 2-generator $X : f \Rightarrow g$, i.e. $(E[f \Rightarrow g])_2 = E_2 \uplus \{X\}$, such that $s_1(X) = f$ and $t_1(X) = g$. The 2-cells in $E[f \Rightarrow g]^*$ being obtained by tensoring and composing 2-generators in $E_2 \uplus \{X\}$, they can be seen as terms in E^* with a metavariable X of type $f \Rightarrow g$. In particular, any morphism of E^* can be seen as a morphism of $E[f \Rightarrow g]^*$ which does not use the metavariable and we write $I_{f \Rightarrow g} : E^* \to E[f \Rightarrow g]^*$ for the corresponding inclusion functor.

By the universal property of the free 2-category $E[f \Rightarrow g]^*$ over the polygraph $E[f \Rightarrow g]$, for any 2-category $\mathcal{D}$, functor $F : E^* \to \mathcal{D}$ and 2-cell $\alpha : F(f) \Rightarrow F(g)$ of $\mathcal{D}$, there exists a unique functor $F[\alpha] : E[f \Rightarrow g]^* \to \mathcal{D}$ such that

$$F[\alpha] \circ I_{f \Rightarrow g} = F \qquad \text{and} \qquad F(X) = \alpha$$

where X is the newly added metavariable. A nice understanding of this can be given by adopting a more abstract definition of polygraphs [14, 31], that we did not give here for the sake of simplicity. As a particular case, *substitution* can be defined using this property: given a 2-cell $\alpha : f \Rightarrow g$ in E_2^*, when $\mathcal{D} = E^*$ and $F : E^* \to E^*$ is the identity functor Id, the image of the image $\mathrm{Id}[\alpha](\beta)$ of a 2-cell $\beta : h \Rightarrow i$ of $E_2[f \Rightarrow g]^*$ is denoted by $\beta[\alpha]$, and corresponds to the morphism obtained from β by replacing every instance of the metavariable X by α.

We more generally write $E[f_1 \Rightarrow g_1, \ldots, f_n \Rightarrow g_n]$ for $(((E[f_1 \Rightarrow g_1]) \ldots)[f_n \Rightarrow g_n])$, extend notation for substitution accordingly, and often write X_i for the newly introduced variable of type $f_i \Rightarrow g_i$. If we fix an enumeration $E_2 = \{\alpha_1, \ldots, \alpha_n\}$ of the 2-generators of E, and write f_i (resp. g_i) for the source (resp. target) of α_i, then the polygraph E is isomorphic to $E'[f_1 \Rightarrow g_1, \ldots, f_n \Rightarrow g_n]$, where E' is the 2-polygraph obtained from E by removing all the 2-generators, i.e. $E_2' = \emptyset$. Now, given a fixed index k, we write $\phi : E_2 \to \mathbb{N}$ for the function such that $\varphi(\alpha_i) = 0$ if $i \neq k$, and $\varphi(\alpha_k) = 1$. If we write $\mathcal{N}$ for the 2-category with one 0-cell, one 1-cell, and $\mathbb{N}$ as set of 2-cells with addition as vertical composition and 0 as vertical identity, and $F : E'^* \to \mathcal{N}$ for the only functor between the

2-categories E'^* and $\mathcal{N}$, the deduced functor $F[\varphi] : E^* \to \mathcal{N}$ sends a 2-cell β to an integer, which is called the *weight* of the 2-generator α_k in β and written $\|\beta\|_{\alpha_k}$. The weight $\|\beta\|_\alpha$ indicates the number of times a 2-generator α of E occurs in a 2-cell β of E^*. Similarly, the *size* $\|\beta\|$ of a 2-cell β is the number of 2-generators that it contains, i.e. $\|\beta\| = \sum_{\alpha \in E_2} \|\beta\|_\alpha$.

Definition 3.2. The *multicategory of contexts* of a 2-polygraph E, denoted by $\mathcal{K}_E$, is the symmetric multicategory whose

    – objects are pairs of parallel 1-cells f and g of E^*, which are often written $f \Rightarrow g$,
    – operations $\alpha : f_1 \Rightarrow g_1 , \ \ldots \ , \ f_n \Rightarrow g_n \to f \Rightarrow g$ are the 2-cells α of type $f \Rightarrow g$ in $E[f_1 \Rightarrow g_1, \ldots, f_n \Rightarrow g_n]^*$ such that for every index i, $\|\alpha\|_{X_i} = 1$, i.e. every metavariable occurs exactly once in the morphism,

composition is induced by substitution as expected, the identity $\mathrm{id}_{f \Rightarrow g} : f \Rightarrow g \to f \Rightarrow g$ on $f \Rightarrow g$ is the variable $X : f \Rightarrow g$, and symmetry corresponds to renaming of variables.

Example 3.3. If we consider the 2-polygraph E of symmetries introduced in Example 2.19, the morphism

$$(\gamma \otimes \mathrm{id}_1) \circ (\mathrm{id}_1 \otimes \gamma) \circ (\mathrm{id}_1 \otimes (X_2 \circ X_1) \otimes \mathrm{id}_1) \circ (\mathrm{id}_1 \otimes \gamma) \circ (\gamma \otimes \mathrm{id}_1)$$

in $E[1 \Rightarrow 1, 1 \Rightarrow 1]$ is a morphism in $\mathcal{K}_E(1 \Rightarrow 1, 1 \Rightarrow 1; 3 \Rightarrow 3)$. Graphically,
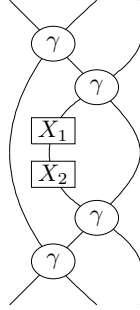

In particular, if we restrict to unary operations (operations whose type is of the form $f_1 \Rightarrow g_1 \to f \Rightarrow g$) the structure of multicategory reduces to a structure of category of contexts, which acts on the set of 2-cells of the 2-category E^*: if $K : f_1 \Rightarrow g_1 \to f \Rightarrow g$ is a unary context, and $\alpha : f_1 \Rightarrow g_1$ is a 2-cell of $\mathcal{C}$, we often write $K(\alpha) : f \Rightarrow g$ for the 2-cell $K[\alpha]$ of $\mathcal{C}$ obtained from K by substituting the variable by α. More generally, a context in a 3-polygraph is a context in the underlying 2-polygraph, and if $K : f_1 \Rightarrow g_1 \to f \Rightarrow g$ is a context in a 3-polygraph and $r : \alpha \Rrightarrow \beta : f_1 \Rightarrow g_1$ is a 3-cell in the 3-category E^*, we write $K(r) : K(\alpha) \Rrightarrow K(\beta) : f \Rightarrow g$ for the obvious 3-cell obtained from r by composing (in dimensions 0 and 1) with identity 2-cells. Notice also that the nullary operations are precisely the 2-cells of E^*.

It is possible to more generally define a notion of multicategory of contexts $\mathcal{K}_\mathcal{C}$ of any 2-category $\mathcal{C}$, which coincides with the previously given definition in the case where $\mathcal{C}$ is of the form $\mathcal{C} = E^*$ for some 2-polygraph E. We will only need to consider this last case in the following, which is why we do not give the general definition.

3.2. **Critical pairs in 3-polygraphs.** Suppose fixed a 3-polygraph E, freely generating a 3-category E^*. Two coinitial 3-cells

$$r_1 : \alpha \Rrightarrow \beta_1 : f \Rightarrow g : A \to B \quad \text{and} \quad r_2 : \alpha \Rrightarrow \beta_2 : f \Rightarrow g : A \to B$$

of this 3-category are *joinable* when there exists a 2-cell $\beta : f \Rightarrow g$ and two 3-cells $s_1 : \beta_1 \Rrightarrow \beta$ and $s_2 : \beta_2 \Rrightarrow \beta$ such that $s_1 \circ_2 r_1 = s_2 \circ_2 r_2$, where $\circ_2$ denotes the composition in dimension 2:

$$
\begin{array}{ccc}
 & \alpha & \\
r_1 \swarrow & & \searrow r_2 \\
\beta_1 & & \beta_2 \\
s_1 \searrow & & \swarrow s_2 \\
 & \beta &
\end{array}
$$
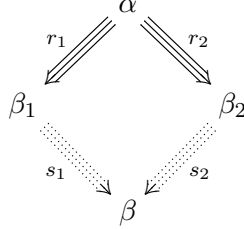
Given a 3-generator $r : \alpha \Rrightarrow \beta$ and two 2-cells α' and β', we write $\alpha' \Rrightarrow^{K,r} \beta'$, when there exists a unary context K such that $K(\alpha) = \alpha'$ and $K(\beta) = \beta'$. A polygraph is *locally confluent* when for every cells such that $\alpha \Rrightarrow^{K_1,r_1} \beta_1$ and $\alpha \Rrightarrow^{K_2,r_2} \beta_2$, the two 3-cells $K_1(r_1)$ and $K_2(r_2)$ are joinable. It is *terminating* when there is no infinite sequence $\alpha_1 \Rrightarrow^{K_1,r_1} \alpha_2 \Rrightarrow^{K_2,r_2} \ldots$

The Newman lemma is valid in this framework [14]:

Lemma 3.4. *A terminating polygraph is confluent if and only if it is locally confluent.*

In some simple cases, termination of polygraphs can be deduced from the fact that all the rules make the size of morphisms decrease:

Lemma 3.5. *If E is a 3-polygraph such that for every 3-generator $r : \alpha \Rrightarrow \beta$ we have $\|\alpha\| > \|\beta\|$, then E is terminating.*

This simple criterion for showing the termination of a polygraph is often too weak. More elaborate termination orders for 3-polygraphs have been studied by Guiraud [11]. In this paper, we are mostly interested in studying local confluence of polygraphs and will not detail those. The usual notion of critical pair can be extended to the setting of 3-polygraphs as follows.

Definition 3.6. A *unifier* of a pair of 2-cells

$$\alpha_1 : f_1 \Rightarrow g_1 \quad \text{and} \quad \alpha_2 : f_2 \Rightarrow g_2$$

in a 2-category $\mathcal{C}$ consists of a pair of cofinal unary contexts

$$K_1 : f_1 \Rightarrow g_1 \to f \Rightarrow g \quad \text{and} \quad K_2 : f_2 \Rightarrow g_2 \to f \Rightarrow g$$

such that $K_1(\alpha_1) = K_2(\alpha_2)$. A unifier is *most general* when it is
- *non-trivial*: there is no binary context

$$K \quad : \quad f_1 \Rightarrow g_1, f_2 \Rightarrow g_2 \quad \to \quad f \Rightarrow h$$

such that

$$K_1 = K \circ (\text{id}_{f_1 \Rightarrow g_1}, \alpha_2) \quad \text{and} \quad K_2 = K \circ (\alpha_1, \text{id}_{f_2 \Rightarrow g_2})$$

- *minimal*: for every unifier (K_1', K_2') of α_1 and α_2 such that $K_1 = K_1'' \circ K_1'$ and $K_2 = K_2'' \circ K_2'$ for some contexts K_1'' and K_2'', the unary contexts K_1'' and K_2'' are invertible.

Definition 3.7. A *critical pair* (K_1, r_1, K_2, r_2) in a 3-polygraph S consists of a pair of 3-generators

$$r_1 : \alpha_1 \Rrightarrow \beta_1 : f_1 \Rightarrow g_1 \quad \text{and} \quad r_2 : \alpha_2 \Rrightarrow \beta_2 : f_2 \Rightarrow g_2$$
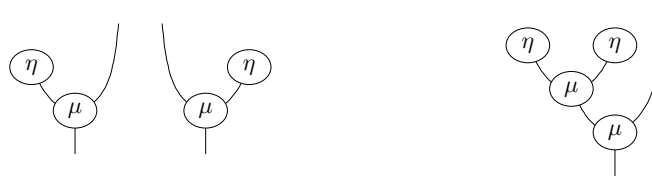
and a most general unifier

$$K_1 : f_1 \Rightarrow g_1 \to f \Rightarrow g \quad \text{and} \quad K_2 : f_2 \Rightarrow g_2 \to f \Rightarrow g$$

of α_1 and α_2. We sometimes say that the 2-cell $\alpha = K_1(\alpha_1) = K_2(\alpha_2)$ is a critical pair, by abuse of language.

Example 3.8. Consider the 3-polygraph E presenting the simplicial category Δ introduced in Example 2.18. The morphisms

are unifiers of the rules l and r which are not most general because they are respectively trivial and not minimal. The critical pairs (and thus the most general unifiers) have been described in Example 2.18.

3.3. **Compact 2-categories.** As explained in the introduction of the present section, our aim is intuitively to be able to consider diagrams such as the one on the left of (3.1) as morphisms. Here, we achieve this by embedding the 2-category E^* generated by a polygraph E into the compact 2-category it freely generates. In these categories, every 1-cell admits both a left and right adjoint, which graphically essentially amounts to have the possibility to bend wires as in the right of (3.1).

The notion of adjunction in the 2-category **Cat** of categories, functors and natural transformations can be generalized to any 2-category $\mathcal{C}$ as follows [21]. A 1-cell $f : A \to B$ is *left adjoint* to a 1-cell $g : B \to A$ (or g is *right adjoint* to f), that we will write $f \dashv g$, when there exist two 2-cells $\eta : \mathrm{id}_A \Rightarrow f \otimes g$ and $\varepsilon : g \otimes f \Rightarrow \mathrm{id}_B$, called respectively the *unit* and the *counit* of the adjunction and depicted respectively by

such that $(f \otimes \varepsilon) \circ (\eta \otimes f) = \mathrm{id}_f$ and $(\varepsilon \otimes g) \circ (g \otimes \eta) = \mathrm{id}_g$. These equations are called the *zig-zag laws* because of their graphical representation:

The notion of 2-category with adjoints was studied in the case of symmetric monoidal categories [20] (where they are called *compact closed* categories), monoidal categories [18] (where they are called *autonomous* categories), as well as other variants such as *spherical* categories [3]; see [34] for a concise presentation of those.

Definition 3.9. A 2-category is *compact* when every 1-cell admits both a left and a right adjoint. A *strictly compact* 2-category is a compact 2-category in which every 1-cell $f : A \to B$ has an assigned left adjoint $f^{-1} : B \to A$ and an assigned right adjoint $f^{+1} : B \to A$. We write η_f^+ and ε_f^+ (resp. η_f^- and ε_f^-) for the unit and the counit of the adjunction $f \dashv f^{+1}$ (resp. $f^{-1} \dashv f$). The following coherence axioms should moreover be satisfied:

    – for every pair of composable 1-cells f and g,

$$(f \otimes g)^{-1} = g^{-1} \otimes f^{-1} \qquad\qquad (f \otimes g)^{+1} = g^{+1} \otimes f^{+1}$$

and

$$\eta_{f\otimes g}^+ = (f \otimes \eta_g^+ \otimes f^{+1}) \circ \eta_f^+ \qquad\qquad \varepsilon_{f\otimes g}^+ = \varepsilon_g^+ \circ (g^{+1} \otimes \varepsilon_f^+ \otimes g)$$

and

$$\eta_{f\otimes g}^- = (g^{-1} \otimes \eta_f^- \otimes g) \circ \eta_g^- \qquad\qquad \varepsilon_{f\otimes g}^- = \varepsilon_f^- \circ (f \otimes \varepsilon_g^- \otimes f^{-1})$$

    – for every 0-cell A,

$$\mathrm{id}_A^{-1} = \mathrm{id}_A = \mathrm{id}_A^{+1}$$

and

$$\eta_{\mathrm{id}_A}^+ = \mathrm{id}_A = \varepsilon_{\mathrm{id}_A}^+ \qquad\qquad \eta_{\mathrm{id}_A}^- = \mathrm{id}_A = \varepsilon_{\mathrm{id}_A}^-$$

    – for every 1-cell f,

$$(f^{+1})^{-1} = f = (f^{-1})^{+1}$$

and

$$\eta_{f^{-1}}^+ = \eta_f^- \qquad\qquad \varepsilon_{f^{-1}}^+ = \varepsilon_f^- \qquad\qquad \eta_{f^{+1}}^- = \eta_f^+ \qquad\qquad \varepsilon_{f^{+1}}^- = \varepsilon_f^+$$

For any 1-cell $f : A \to B$ in a strictly compact 2-category and integer n, the morphism f^n denotes the morphism defined by $f^0 = f$, $f^{n+1} = (f^n)^{+1}$ and $f^{n-1} = (f^n)^{-1}$. We also simply write $\eta_f : B \Rightarrow f^{-1} \otimes f$ and $\varepsilon_f : f \otimes f^{-1} \Rightarrow A$ for the unit and the counit of the adjunction between f^{-1} and f.

In the following, we suppose for simplicity that all the compact categories we consider are equipped with a structure of strictly compact category. This is not restrictive since every compact 2-category can be shown to be equivalent to a strict one using an argument similar to the coherence theorem for compact closed categories [20]. The category of compact categories is denoted by $\mathbf{CCat}_2$.

3.4. **Embedding 2-categories into compact 2-categories.** There is an obvious forgetful functor from the category of compact 2-categories to the category of 2-categories, and this forgetful functor admits a left adjoint. We write $\mathcal{A}_{\mathcal{C}}$ for the free compact 2-category on a 2-category $\mathcal{C}$ (the $\mathcal{A}$ here stands for "adjoints"). The construction of this free 2-category is detailed in [33] and consists essentially in adapting the work of Kelly and Laplaza on compact closed categories [20] to monoidal categories which are not supposed to be symmetric. We recall briefly this construction here.

The underlying category of a compact 2-category is naturally equipped with a structure of "category with formal adjoints" in the following sense:

Definition 3.10. A *category with formal adjoints* $(\mathcal{C}, (-)^{-1}, (-)^{+1})$ is a category together with two functors

$$(-)^{-1} : \mathcal{C} \to \mathcal{C}^{\mathrm{op}} \quad \text{and} \quad (-)^{+1} : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$$

such that $((-)^{-1})^{+1} = \mathrm{id}_{\mathcal{C}}$ and $((-)^{+1})^{-1} = \mathrm{id}_{\mathcal{C}^{\mathrm{op}}}$.

Given a 2-category $\mathcal{C}$, with $\mathcal{C}|_1$ as underlying category, the underlying category of $\mathcal{A}_{\mathcal{C}}$ is the free category with formal adjoints on $\mathcal{C}|_1$. More concretely, this category is the free category on the graph whose objects are the objects of $\mathcal{C}|_1$ as objects and whose arrows $f^n : A \to B$ are pairs constituted of an integer $n \in \mathbb{Z}$, called a *winding number*, and a morphism $f : A \to B$ in $\mathcal{C}$ if n is even (resp. a morphism $f : B \to A$ in $\mathcal{C}$ if n is odd), quotiented by the following equalities:

– for every pair of composable morphisms f^n and g^n,

$$f^n \otimes g^n \quad = \quad \begin{cases} (f \otimes g)^n & \text{if } n \text{ is even} \\ (g \otimes f)^n & \text{if } n \text{ is odd} \end{cases}$$

– for every object A,

$$(\mathrm{id}_A)^n \quad = \quad \mathrm{id}_A$$

The 2-cells of $\mathcal{A}_{\mathcal{C}}$ are formal vertical and horizontal composites of

– $\alpha^0 : f^0 \Rightarrow g^0$, where $\alpha : f \Rightarrow g$ is a 2-cell of $\mathcal{C}$,
– $\eta_{f^n} : \mathrm{id}_B \Rightarrow f^{n-1} \otimes f^n$, for every 1-cell $f^n : A \to B$,
– $\varepsilon_{f^n} : f^n \otimes f^{n-1} \Rightarrow \mathrm{id}_A$, for every 1-cell $f^n : A \to B$,

quotiented by

– the axioms of 2-categories (see Definition 1.3),
– for every pair of vertically composable 2-cells α^0 and β^0,

$$\beta^0 \circ \alpha^0 \quad = \quad (\beta \circ \alpha)^0$$

– for every 1-cell f^0,

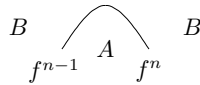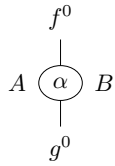$$\mathrm{id}_{f^0} \quad = \quad (\mathrm{id}_f)^0$$

– for every pair of horizontally composable 2-cells α^0 and β^0,

$$\alpha^0 \otimes \beta^0 \quad = \quad (\alpha \otimes \beta)^0$$

– for every 1-cell f^n,

$$(f^{n-1} \otimes \varepsilon_{f^n}) \circ (\eta_{f^n} \otimes f^{n-1}) = f^{n-1} \quad \text{and} \quad (\varepsilon_{f^n} \otimes f^n) \circ (f^n \otimes \eta_{f^n}) = f^n$$

Graphically, if we write respectively
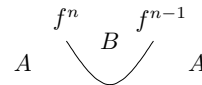
for $\alpha^0 : f^0 \Rightarrow g^0 : A \to B$, η_{f^n} and ε_{f^n} (where $f^n : A \to B$), the four last equalities can be pictured as follows

The string diagrams for compact categories are studied in [19].

Remark 3.11. If $\mathcal{C}$ is the 2-category E^* generated by a 2-polygraph E, the compact 2-category $\mathcal{A}_{\mathcal{C}}$ is presented by the 3-polygraph F such that

- $F_0 = E_0$
- $F_1 = \{\, f^n \mid f \in E_1, n \in \mathbb{Z} \,\}$
- $F_2 = \{\, \alpha^0 \mid \alpha \in E_2 \,\} \uplus \{\, \eta_{f^n}, \varepsilon_{f^n} \mid f^n \in F_1 \,\}$
- $F_3 = \{\, l_{f^n}, r_{f^n} \mid f^n \in F_1 \,\}$

with

$$
\begin{aligned}
l_{f^n} &: & (f^{n-1} \otimes \varepsilon_{f^n}) \circ (\eta_{f^n} \otimes f^{n-1}) &\Rightarrow & f^{n-1} \\
r_{f^n} &: & (\varepsilon_{f^n} \otimes f^n) \circ (f^n \otimes \eta_{f^n}) &\Rightarrow & f^n
\end{aligned}
$$

and other cells have the obvious source and target. By Lemma 3.5, the polygraph F is terminating and by Lemma 3.4 it is confluent since all its critical pairs, which are of the form

 and

for some 1-cell f^n, are joinable.

Lemma 3.12. *With the notations of the preceding remark, if $f_1, \ldots, f_m$ and $g_1, \ldots, g_n$ are parallel lists of composable morphisms of E^*, then the 2-cells*

$$\alpha \quad : \quad f_1^0 \otimes \ldots \otimes f_m^0 \quad \Rightarrow \quad g_1^0 \otimes \ldots \otimes g_n^0$$

in the underlying 2-category of F which are normal forms (with respect to the rewriting rules of F) do not contain any 2-generator η_{f^k} or ε_{f^k}.

Proof. It is easy to show that a 2-cell α in F^* can be written as a composite of morphisms of the form $\mathrm{id}_g \otimes \beta \otimes \mathrm{id}_h$ where β is either a 2-cell of $\mathcal{C}$ or a morphism of the form η_{f^k} or ε_{f^k} (see for example [22]). Suppose that α contains a 2-generator of the form ε_{f^k} with $k > 0$. It can therefore be written as a composite of the form
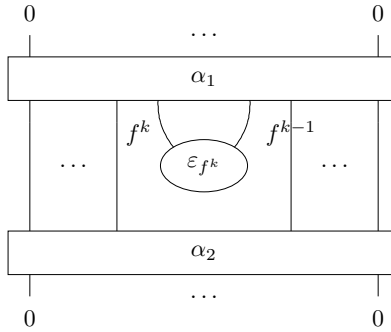
The 2-cells α_1 and α_2 are noted with boxes for clarity and 0 stands for a 1-cell whose winding number is 0. Since the only generators whose target contain a 1-cell of the form f^k are η_k and η_{k+1}, the 2-cell α_1 is necessarily of one of the three following forms:
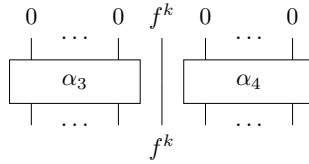
or
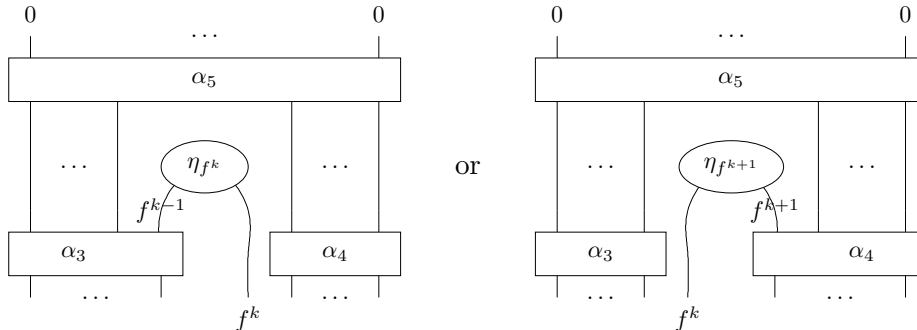
The first case is impossible since $k > 0$ and we have supposed that all the winding numbers of the 1-generators occurring in the source of α are 0. The second case is not possible either

since the morphism α would not be a normal form (the rule $l_{f^{k-1}}$ could be applied), and the 2-cell α thus contains a 2-generator $\eta_{f^{k+1}}$. By using a similar argument, α also contains the 2-generator $\varepsilon_{f^{k+2}}$. So, by induction, the 2-cell α would contain all the 2-generators $\varepsilon_{f^{k+2i}}$ with $i \in \mathbb{N}$ and would therefore be a composite of an infinite number of generators. This is absurd since the 2-cells in F^* are inductively generated. We deduce that α does not contain a 2-generator of the form ε_{f^k} with $k > 0$. Similarly, it does not contain a 2-generator η_{f^k} with $k > 0$. The cases where $k \leqslant 0$ are also similar (we construct an infinite sequence of generators that α would contain, with strictly decreasing winding numbers). $\qquad\square$

From this, we can deduce that the 2-cells

$$\alpha \quad : \quad f_1^0 \otimes \ldots \otimes f_m^0 \quad \Rightarrow \quad g_1^0 \otimes \ldots \otimes g_n^0$$

in F^* are in bijection with the 2-cells

$$\alpha \quad : \quad f_1 \otimes \ldots \otimes f_m \quad \Rightarrow \quad g_1 \otimes \ldots \otimes g_n$$

of $\mathcal{C}$, which shows that the embedding of E^* into F^* is full and faithful (this embedding is the functor defined as the identity on objects, as $f \mapsto f^0$ on 1-cells and as $\alpha \mapsto \alpha^0$ on 2-cells). A 2-cell α whose source and target is of the form above is called *regular* (the regular 2-cells are thus those which are in the image of the embedding). The argument can easily be generalized to any category $\mathcal{C}$, not necessarily generated by a 2-polygraph (but we will only make use of the case proved in previous lemma):

Proposition 3.13. *The components $\eta_{\mathcal{C}} : \mathcal{C} \to \mathcal{A}_{\mathcal{C}}$ of the unit of the adjunction between 2-categories and compact 2-categories are full and faithful.*

This means that given two 0-cells A and B of $\mathcal{C}$, the hom-categories $\mathcal{C}(A, B)$ and $\mathcal{A}_{\mathcal{C}}(A, B)$ are isomorphic in a coherent way. The 2-category $\mathcal{A}_{\mathcal{C}}$ thus provides a "larger world" in which we can embed the 2-category $\mathcal{C}$ without losing information.

An interesting observation about compact 2-categories is that, in those, the distinction between the source and the target of a 1-cell is quite "artificial". This is formalized by the following proposition.
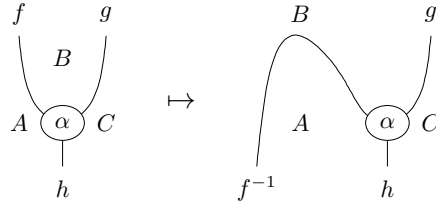
Proposition 3.14. *If $\mathcal{C}$ is a compact 2-category, the sets*

$$\mathrm{Hom}(f \otimes g, h) \quad \cong \quad \mathrm{Hom}(g, f^{-1} \otimes h)$$

are naturally isomorphic by the function

$$\alpha \quad \mapsto \quad (f^{-1} \otimes \alpha) \circ (\eta_f \otimes g)$$

Graphically,

Similarly, the sets

$$\mathrm{Hom}(f \otimes g, h) \quad \cong \quad \mathrm{Hom}(f, h \otimes g^1)$$

are naturally isomorphic by the function

$$\alpha \quad \mapsto \quad (\alpha \otimes g^1) \circ (f \otimes \eta_{g^1})$$

These bijections are called rotations.

In particular, for any pair of 1-cells $f, g : A \to B$, the set $\mathrm{Hom}(f, g)$ is isomorphic to $\mathrm{Hom}(\mathrm{id}_B, f^{-1} \otimes g)$ and to $\mathrm{Hom}(f \otimes g^{-1}, \mathrm{id}_A)$. This shows that the notion of "input" and "output" of 2-cells is fairly artificial in compact 2-categories. Actually, based on these ideas, it is possible to reformulate equivalently the notion of compact 2-category in a way such that 2-cells only have one "border" instead of having both a source and a target. We have called the resulting notion a *rotative 2-category*, details can be found in [31].

Proposition 3.13 provides us with a full and faithful embedding $\mathcal{C} \to \mathcal{A}_\mathcal{C}$ of a 2-category $\mathcal{C}$ into the compact 2-category that it freely generates. The interest of this embedding is that there are "extra morphisms" in $\mathcal{A}_\mathcal{C}$ that can be used to represent "partial compositions" in $\mathcal{C}$. For example, consider two 2-cells $\alpha : f \Rightarrow f_1 \otimes g \otimes f_2$ and $\beta : h_1 \otimes g \otimes h_2 \Rightarrow h$ in $\mathcal{C}$. These can be seen as the morphisms of $\mathcal{A}_\mathcal{C}$ depicted on the left of (3.2) by the previous embedding.

$$(3.2)$$
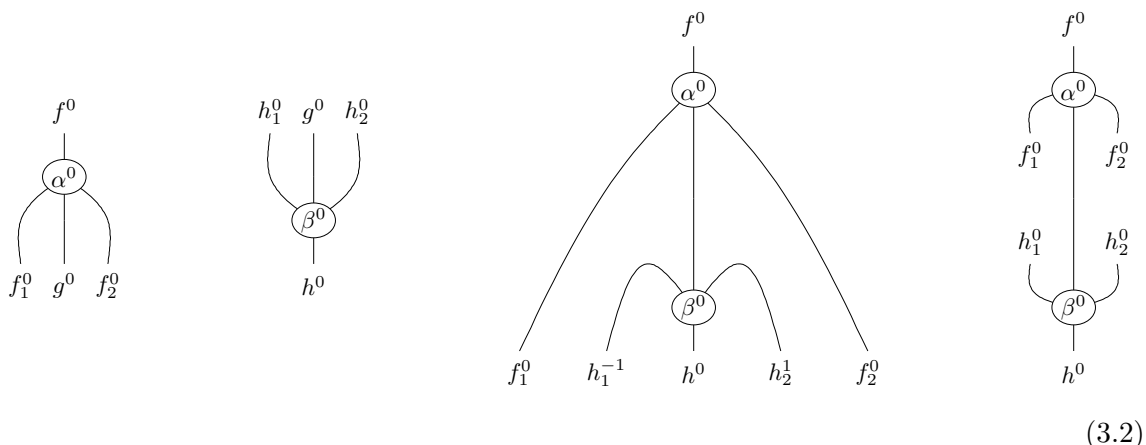
From these two morphisms, the morphism $\alpha \otimes_g \beta : f^0 \Rightarrow f_1^0 \otimes h_1^{-1} \otimes h^0 \otimes h_2^1 \otimes f_2^0$, depicted in the center right of (3.2), can be constructed. This morphism represents the *partial composition* of the 2-cells α and β on the 1-cell g: up to rotations, this 2-cell is fundamentally a way to give a precise meaning to the diagram depicted on the right of (3.2).

3.5. **Compact polygraphs.** The notion of *compact 2-polygraph E* is defined as in (1.13), where E_1^* is now the set of morphisms of the free category with formal adjoints on the graph (1.1), and E_2^* is the set of 2-cells of the free compact 2-category with the previously generated category as underlying category with formal adjoints. We write **CPol$_2$** for the category of compact 2-polygraphs.

Similarly, the construction of the multicategory of contexts given in Section 3.1 can be straightforwardly adapted to compact polygraphs. Given a compact 2-polygraph E, we still write $\mathcal{K}_E$ for the multicategory thus generated, whose operations are then called *compact contexts*. The setting of compact contexts provides a generalization of partial composition by allowing a "partial composition of a morphism with itself". Namely, from a context

$\alpha : (\ldots, (f_i, g_i), \ldots) \to (f, g^1 \otimes h \otimes g^0)$ with $f : A \to A$ and $h : B \to B$ one can build the context

$$\varepsilon_g^0 \circ (g^1 \otimes X \otimes g^0) \circ \alpha \quad : \quad (\ldots, (f_i, g_i), \ldots, (h, \mathrm{id}_B)) \quad \to \quad (f, \mathrm{id}_A)$$

where $X : h \to \mathrm{id}_B$ is a fresh variable. Graphically,
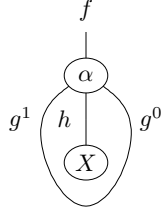
This operation amounts to *merging* the outputs of type g^1 and g^0 of α, thus creating a "hole" which is formally taken in account as a metavariable in the multicategory of compact contexts.

3.6. **Towards a unification algorithm for 3-polygraphs.** Using the partial composition and merging operations defined above, it is possible to formulate an algorithm for computing critical pairs in a 3-polygraph, which will proceed as follows. Suppose given a 3-polygraph E. This polygraph generates a 3-category E^* whose underlying 2-category is written $\mathcal{C}$ (the 2-category $\mathcal{C}$ is freely generated by the underlying 2-polygraph $E|_2$ of E). The 2-category $\mathcal{C}$ can be fully and faithfully embedded into the free compact 2-category $\mathcal{A}_{\mathcal{C}}$ it generates (Proposition 3.13). In turn, this compact 2-category generates a multicategory of contexts $\mathcal{K}_{\mathcal{A}_{\mathcal{C}}}$ in which it can be embedded (any 2-cell of $\mathcal{A}_{\mathcal{C}}$ can be seen as a nullary context in $\mathcal{K}_{\mathcal{A}_{\mathcal{C}}}$). Given two rewriting rules

$$r' : \alpha' \Rrightarrow \beta' : f' \Rightarrow g' \qquad \text{and} \qquad r'' : \alpha'' \Rrightarrow \beta'' : f'' \Rightarrow g''$$

the two 2-cells $\alpha' : f' \Rightarrow g'$ and $\alpha'' : f'' \Rightarrow g''$ in $\mathcal{C}$ can be seen as 2-cells $\alpha'^0 : f'^0 \Rightarrow g'^0$ and $\alpha''^0 : f''^0 \Rightarrow g''^0$ in $\mathcal{A}_{\mathcal{C}}$, which in turn can be seen as nullary contexts in $\mathcal{K}_{\mathcal{A}_{\mathcal{C}}}(; f'^0 \Rightarrow g'^0)$ and $\mathcal{K}_{\mathcal{A}_{\mathcal{C}}}(; f''^0 \Rightarrow g''^0)$ respectively. Our algorithm will compute a unifier of those in the category of compact contexts, consisting of a pair

$$K' \quad : \quad f'^0 \Rightarrow g'^0, f_1 \Rightarrow g_1, \ldots, f_n \Rightarrow g_n \quad \to \quad f \Rightarrow g$$

and

$$K'' \quad : \quad f''^0 \Rightarrow g''^0, f_1 \Rightarrow g_1, \ldots, f_n \Rightarrow g_n \quad \to \quad f \Rightarrow g$$

of compact contexts such that

$$K'(\alpha'^0, \mathrm{id}_{f_1 \Rightarrow g_1}, \ldots, \mathrm{id}_{f_n \Rightarrow g_n}) \quad = \quad K''(\alpha''^0, \mathrm{id}_{f_1 \Rightarrow g_1}, \ldots, \mathrm{id}_{f_n \Rightarrow g_n}) \tag{3.3}$$

which is minimal and non-trivial (in a sense similar to Definition 3.6), up to the symmetry of the multicategory and rotations. Notice that the usual critical pairs in 3-polygraphs having 2-generators of arity 1 (e.g. the polygraph of monoids given in Example 2.18) are recovered as the particular case where the unifiers are such that $n = 0$ (there is no hole in the unifier) and the morphism (3.3) is regular, i.e. the morphism (3.3) is in the image of the embedding of $\mathcal{C}$ into $\mathcal{K}_{\mathcal{A}_{\mathcal{C}}}$. When it is not the case, all the unifiers of α' and α'' (in the sense of Definition 3.6) can be recovered as the contexts of the form

$$K \circ K'(\mathrm{id}_{f' \Rightarrow g'}, \alpha_1, \ldots, \alpha_n) \qquad \text{and} \qquad K \circ K'(\mathrm{id}_{f'' \Rightarrow g''}, \alpha_1, \ldots, \alpha_n)$$

where the α_i are morphisms in $\mathcal{A}_\mathcal{C}$ seen as nullary compact contexts and K is a unary regular compact context. In this sense, the critical pairs in the category of compact contexts generate all the unifiers in the usual sense, and have the advantage of always being finite in number for a finite polygraph.

A concrete description of the algorithm is out of the scope of this paper since it requires the elaboration of an explicit representation of the morphisms in the 2-category E^* generated by a 2-polygraph E [30, 31]: up to now we have defined these morphisms using an abstract universal construction (see Section 1.2.3) however a more concrete representation is needed in order manipulate them algorithmically. Such a representation was developed by the author, by describing the 2-cells in E^* themselves as polygraphs labeled by E (i.e. objects in the slice category $\mathbf{Pol}_2 \downarrow E$) up to isomorphism. For example, if E is the 2-polygraph corresponding to the signature of monoids defined in Example 2.18, the morphism $\mu \circ (\mu \otimes \mathrm{id}_1)$ can be represented by the polygraph M such that

$$M_0 = \{1,\ 2,\ 3,\ 4\} \qquad M_1 = \{5 : 1 \to 2,\ 6 : 2 \to 3,\ 7 : 3 \to 4,\ 8 : 1 \to 3,\ 9 : 1 \to 4\}$$

and

$$M_2 = \{10 : 5 \otimes 6 \Rightarrow 8,\ 11 : 8 \otimes 7 \Rightarrow 9\}$$

(we have chosen to number the generators in an arbitrary order) labeled by the morphism of polygraphs $\lambda : M \to E$ such that all the 0-, 1- and 2-generators are labeled by $*$, 1 and μ respectively. Graphically, this corresponds to give different names (or numbers in this case) to the various instances of generators used to build the morphism:
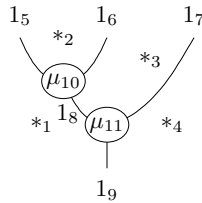

Of course, the naming of generators is not canonical which explains why we have to consider these labeled polygraphs up to isomorphism (this can be seen as some form of α-conversion). The precise description of this 2-category can be achieved by constructing a structure of monoidal globular category [4] on the globular category of polygraphs. This will be presented in detail in further work.

We shall only illustrate how our algorithm works, by giving an example. Consider a 3-polygraph E with one 0-generator $*$, one 1-generator $1 : * \to *$, three 2-cells $\delta : 1 \to 4$, $\mu : 4 \to 1$ and $\sigma : 1 \to 1$ (where 4 denotes $1 \otimes 1 \otimes 1 \otimes 1$) and two rewriting rules (3-generators) whose left members are respectively $\alpha = \varsigma \circ \delta$ and $\beta = \mu \circ \varsigma$, where $\varsigma = \sigma \otimes \sigma \otimes \sigma \otimes \sigma$, i.e. string diagrammatically

The algorithm for unifying these two morphisms will go on as follows. It is non-deterministic and all the unifiers of the two morphisms (possibly with duplicates) will be obtained as the collection of all the results of non-failed execution branches the algorithm. It starts by choosing a 2-generator in each of the morphisms (the gray-colored ones in the diagram above), and then it will progressively attach new cells to the left one, or link wires together so that it becomes a most general unifier. If the two chosen 2-generators are not the same then the algorithm fails (here it does not because both are σ). Since the σ generator selected on the morphism on the left is linked with a μ generator, the algorithm will start by adding a μ generator to the morphism on the left and doing a partial composition with it, as shown on the left of (3.4). The unification is then propagated: the μ generator in the morphism on the right is connected with four σ generators. The algorithm will non-deterministically choose one which has not already been unified and will propagate the unification. For example, if the third σ from the left is selected, the morphism in the middle of (3.4) might be obtained by adding a new σ generator to the morphism. Non-deterministically, instead of adding a new generator, the algorithm might choose to merge (using the operation described at the end of Section 3.5) two inputs or outputs of the morphisms and the morphism on the right of (3.4) might be obtained as well (in this case a new hole is added to the unifier being constructed).

(3.4)
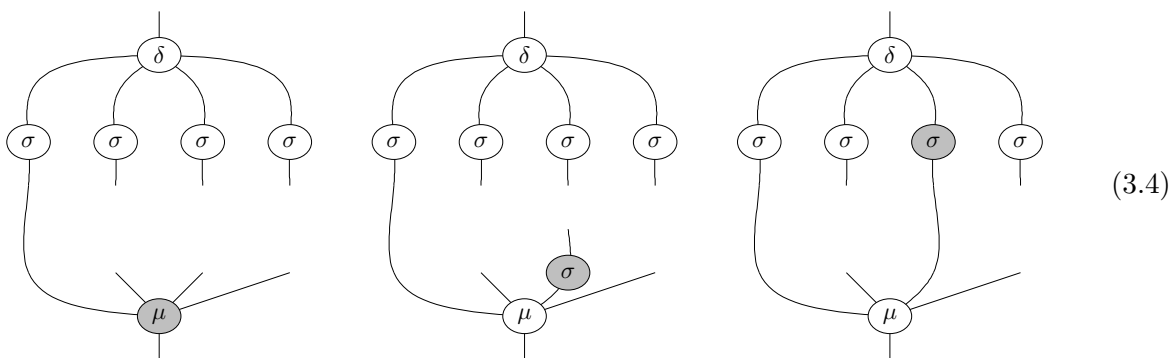
Finally, by fully executing the algorithm, the three morphisms below will be obtained as unifiers (as well as many others).
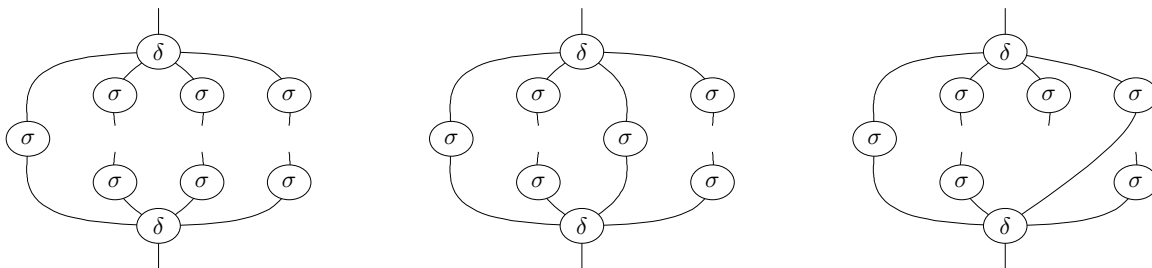
It can be shown that the algorithm terminates and generates all the critical pairs in compact contexts, and these are in finite number. It is important to notice that the algorithm generates the critical pairs of a rewriting system R in the "bigger world" of compact contexts, from which we can generate the critical pairs in the 2-category generated by R (which are not necessarily in finite number as explained in the introduction). If

joinability of the critical pairs in compact contexts implies that the rewriting system is confluent, the converse is unfortunately not true: a similar situation is well known in the study of λ-calculus with explicit substitutions, where a rewriting system might be confluent without being confluent on terms with metavariables.

We have done a toy implementation of the algorithm in less than 2000 lines of OCaml, with which we have been able to successfully recover the critical pairs of rewriting systems in [22]. Even though we did not particularly focus on efficiency, the execution times are good, typically less than a second, because the morphisms involved in polygraphic rewriting systems are usually small (but they can generate a large number of critical pairs). We thus have hope to be able to build efficient tools in order to help dealing with large algebraic structures.

4. Future work

We have tried gradually to expose the notion of higher-dimensional rewriting system and to connect it with the well known and well studied special cases of string and term rewriting system. We have also introduced the notion of multicategory of compact contexts generated by a 2-polygraph, which lays the theoretical foundations for unification in polygraphic 2-dimensional rewriting systems. This leaves many research tracks open for future work, some of which are detailed below.

4.1. **A 2-dimensional unification algorithm.** We have hinted in Section 3.6 how the theoretical tools introduced in this article can be used to formulate a unification algorithm for 3-polygraphs. This algorithm will be described and proven correct in detail in subsequent work [31]. In particular, we plan to study the precise links between our algorithm and the usual unification for term rewriting systems, as well as algorithms for (planar) graph rewriting.

4.2. **Compact rewriting systems.** The use of compact 2-categories seems to be very promising, since it provides a bigger world in which unification is simple to handle (there are a finite number of critical pairs in particular). Moreover, left and right members of rules in polygraphic rewriting systems are morphisms in 2-categories, but we can extend the framework to have "compact rewriting rules" whose left and right members are morphisms in compact 2-categories. There is no known finite convergent polygraphic rewriting system presenting the category **Rel** of finite sets and relations [22] (which corresponds to the theory of qualitative bicommutative bialgebras [29]). We conjecture that such a system does not exist. However, we believe that it would be possible to have a finite convergent compact

polygraphic rewriting system containing rules such as
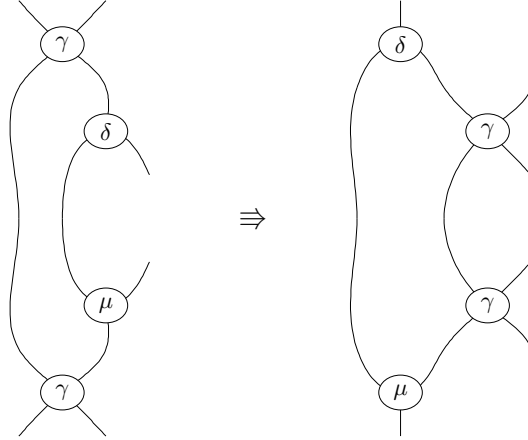
where γ is the generator for the symmetry, δ is the comultiplication and μ is the multiplication. We plan to use our unification algorithm in order to define and study such a rewriting system. It would also be interesting to adapt the techniques developed by Guiraud to show termination of polygraphic systems [11].

4.3. **Parametric polygraphs.** In order to describe those free compact 2-categories, we had to modify the definition of the notion of polygraph by replacing the free category construction by a free category with formal adjoints construction, and the free 2-category construction by a free compact 2-category construction. This suggests that it might be interesting to investigate a more modular notion of polygraph, parametrized by a series of adjunctions, which could be used to generate free n-categories *with properties* (e.g. compact categories, groupoids, etc.).

4.4. **Towards higher dimensions.** Since the notion of polygraphic rewriting system can be generalized to any dimension, we would like to also have a generalization of rewriting theory to higher dimensions using polygraphic rewriting systems. This would require a more abstract and general formulation of the unification techniques that are used here, in order to be able to extend them easily to higher dimensions.

4.5. **Practical uses of this work.** In some sense, our work can be considered as an algebraic study of the notion of a bunch of operators linked by planar wires. We believe that this point of view should be taken seriously and we plan to investigate a possible application of the polygraphic rewriting techniques to electronic circuits. This could provide a nice theoretical framework in which we could express and study optimization of integrated circuits. Another field of application should be the design of an optimizing language for digital signal processing. Sound effects are often described by diagrammatic notation which is very close to the string-diagrammatic notation for morphisms in 2-categories generated

by 2-polygraphs. For instance an echo operator is often pictured as
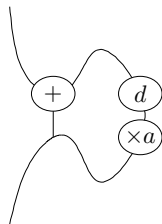
where $+$ adds to signals, $\times a$ amplifies the signal by a coefficient $a < 1$ and d delays the signal for d seconds: the echo is obtained by adding the signal d seconds before, at a lower volume, to the current signal. The rewriting techniques offered by polygraphs could therefore be used in order to optimize those circuits using 3-dimensional rewriting systems.

References

[1] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1999.
[2] J. Baez and M. Stay. Physics, Topology, Logic and Computation: A Rosetta Stone. In *New Structures for Physics*, volume 813 of *Lecture Notes in Physics*, pages 95–172. Springer, 2011.
[3] J. Barrett and B. Westbury. Spherical Categories. *Advances in Mathematics*, 143(2):357–375, 1999.
[4] M. Batanin. Monoidal Globular Categories As a Natural Environment for the Theory of Weak n-Categories. *Advances in Mathematics*, 136(1):39–103, 1998.
[5] A. Burroni. Higher-dimensional word problems with applications to equational logic. *Theoretical Computer Science*, 115(1):43–62, 1993.
[6] M. Fiore, G. Plotkin, and D. Turi. Abstract syntax and variable binding. In *Logic in Computer Science, 1999. Proceedings. 14th Symposium on*, pages 193–202, 2002.
[7] G. Gonthier, J.-J. Lévy, and P.-A. Melliès. An abstract standardisation theorem. In *Proceedings of the 8th Annual Symposium on Logic in Computer Science*, pages 72–81, 1992.
[8] R. Gordon, A. Power, and R. Street. *Coherence for tricategories.* Amer Mathematical Society, 1995.
[9] J. Goubault-Larrecq, S. Lasota, and D. Nowak. Logical relations for monadic types. *Mathematical Structures in Computer Science*, 18(06):1169–1217, 2008.
[10] Y. Guiraud. *Présentations d'opérades et systèmes de réécriture.* PhD thesis, Université Montpellier 2, 2004.
[11] Y. Guiraud. Termination orders for 3-dimensional rewriting. *Journal of Pure and Applied Algebra*, 207(2):341–371, 2006.
[12] Y. Guiraud. The three dimensions of proofs. *Annals of pure and applied logic*, 141(1-2):266–295, 2006.
[13] Y. Guiraud. Two polygraphic presentations of Petri nets. *Theoretical Computer Science*, 360(1-3):124–146, 2006.
[14] Y. Guiraud and P. Malbos. Higher-dimensional categories with finite derivation type. *Theory and Applications of Categories*, 22(18):420–478, 2009.
[15] N. Gurski. *An algebraic theory of tricategories.* PhD thesis, University of Chicago, 2006.
[16] A. Hatcher. *Algebraic Topology.* Cambridge University Press, 2002.
[17] A. Joyal and R. Street. The Geometry of Tensor Calculus, I. *Advances in Mathematics*, 88:55–113, 1991.
[18] A. Joyal and R. Street. Braided tensor categories. *Advances in Mathematics*, 102(1):20–78, 1993.
[19] A. Joyal and R. Street. Planar Diagrams and Tensor Algebra. Unpublished manuscript, available on Ross Street's website, 1998.
[20] G. Kelly and M. Laplaza. Coherence for compact closed categories. *Journal of Pure and Applied Algebra*, 19:193–213, 1980.

[21] G. Kelly and R. Street. Review on the Elements of 2-categories. *Proceedings Sydney Category Theory Seminar*, 73:75–103, 1972.

[22] Y. Lafont. Towards an algebraic theory of Boolean circuits. *Journal of Pure and Applied Algebra*, 184(2-3):257–310, 2003.

[23] Y. Lafont and A. Prouté. Church-Rosser Property and Homology of Monoids. *Mathematical Structures in Computer Science*, 1(3):297–326, 1991.

[24] F. W. Lawvere. *Functorial Semantics of Algebraic Theories and Some Algebraic Problems in the context of Functorial Semantics of Algebraic Theories*. PhD thesis, Columbia University, 1963.

[25] T. Leinster. *Higher Operads, Higher Categories*. Number 298 in London Mathmatics Society Lecture Note Series. Cambridge University Press, 2004.

[26] S. MacLane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer Verlag, 1971.

[27] P.-A. Melliès. Categorical Semantics of Linear Logic. *Panoramas et synthèses-Société mathématique de France*, (27):1–196, 2009.

[28] S. Mimram. *Sémantique des jeux asynchrones et réécriture 2-dimensionnelle*. PhD thesis, PPS, CNRS – Université Paris Diderot, 2008.

[29] S. Mimram. The Structure of First-Order Causality. In *Proceedings of the 24th symposium on Logic in Computer Science*, pages 212–221, 2009.

[30] S. Mimram. Computing critical pairs in 2-dimensional rewriting systems. In C. Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 227–242, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[31] S. Mimram. Representing 2-Dimensional Critical Pairs. Available at `http://www.pps.jussieu.fr/~smimram/docs/mimram_repr2d.pdf`, 2010.

[32] J. Power. An *n*-categorical pasting theorem. In *Category Theory*, volume 1488 of *Lecture Notes in Mathematics*, pages 326–358. Springer, 1991.

[33] A. Preller and J. Lambek. Free compact 2-categories. *Mathematical Structures in Computer Science*, 17(02):309–340, 2007.

[34] P. Selinger. A Survey of Graphical Languages for Monoidal Categories. In *New Structures for Physics*, volume 813 of *Lecture Notes in Physics*, pages 289–355. Springer, 2011.

[35] C. Squier. Word problems and a homological finiteness condition for monoids. *Journal of Pure and Applied Algebra*, 49(1-2):201–217, 1987.

[36] R. Street. Limits indexed by category-valued 2-functors. *Journal of Pure and Applied Algebra*, 8(2):149–181, 1976.

[37] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

CEA, LIST, LMeASI, 91191 Gif-sur-Yvette, France.
E-mail address: `samuel.mimram@cea.fr`